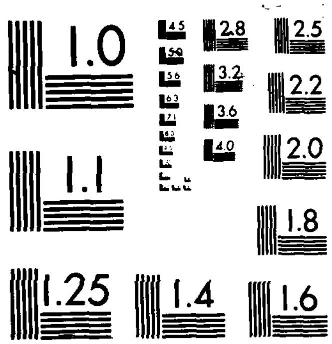
AD-A175 187 LARGE SCALE C3 SYSTEMS EXPERIMENT DESIGN AND SYSTEM IMPROVEMENT(U) MASSACHUSETTS INST OF TECH CAMBRIDGE LAB FOR INFORMATION AND D P J MARTIN AUG 86 LIDS-TH-1580 1/2

UNCLASSIFIED N00014-85-K-0782 F/G 17/2 NL



ROCOPY RESOLUTION TEST CHART

(12)

AD-A175 187

AUGUST 1986

LIDS-TH-1580

Research Supported By:

*Joint Directors of Laboratories
through the Office of
Naval Research
Contract No. N00014-85-K-0782*

LARGE SCALE C³ SYSTEMS: EXPERIMENT DESIGN AND SYSTEM IMPROVEMENT

Philippe J. F. Martin

Laboratory for Information and Decision Systems

MASSACHUSETTS INSTITUTE OF TECHNOLOGY, CAMBRIDGE, MASSACHUSETTS 02139

86 8 28 068

August 1986 LIDS-TH-1580

Large Scale C^3 Systems: Experiment Design and System Improvement

By

Philippe J.F. Martin

This report is based on the unaltered thesis of Philippe J.F. Martin, submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of Master of Science in Technology and Policy at the Massachusetts Institute of Technology in August 1986. The research was conducted at the MIT Laboratory for Information and Decision Systems with support provided by the French Délégation Générale pour l'Armement and with partial support provided by the Joint Directors of Laboratories through the Office of Naval Research under contract no. N00014-85-K-0782.

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA 02139

LARGE SCALE C^3 SYSTEMS:
EXPERIMENT DESIGN AND SYSTEM IMPROVEMENT

By

PHILIPPE J.F. MARTIN

Ingénieur de l'Ecole Polytechnique
(1984)

SUBMITTED TO THE DEPARTMENT OF
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE
IN
TECHNOLOGY AND POLICY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

August 1986

Signature of Author ______________________
Department of Electrical Engineering and Computer Science
August 1, 1986

Certified by ______________________
Alexander H. Levis
Thesis Supervisor

Accepted by ______________________
Richard de Neufville
Chairman, Technology and Policy Program

Accepted by ______________________
Arthur C. Smith
Chairman, departmental Graduate Committee
Electrical Engineering and Computer Science Department

LARGE SCALE C^3 SYSTEMS: EXPERIMENT DESIGN AND SYSTEM IMPROVEMENT

BY

PHILIPPE J.F. MARTIN

Submitted to the department
of Electrical Engineering and Computer Science

on August 1, 1986

in partial fulfillment of the requirements for the degree of
Master of Science in Technology and Policy

ABSTRACT

A methodology for improving a large scale C^3 system in an optimal manner is developed. The first step of this methodology consists of a procedure for designing experiments to run on a large scale C^3 system: it aims at determining the smallest number of experiments that enables one to evaluate the effectiveness of an actual system. The second step consists of a dynamic programming algorithm aimed at determining an optimal sequence of improvements for a system on the basis of the effectiveness analysis.

This methodology is applied to an actual air defense system: each configuration of the system is evaluated with the first step of the methodology; then, the second step is used to find an optimal sequence of modifications for the system as it goes from one configuration to another.

Thesis Supervisor : Dr. Alexander H. Levis
Title: Senior Research Scientist

PER LETTER

A-1

ACKNOWLEDGEMENT

Working with Dr. Alexander H. Levis was a very valuable experience. He introduced me to the world of research: I wish to thank him for his guidance and support. I have very much appreciated working with him.

I would like to also thank Lisa Babine for being so helpful, and Art Giordani of LIDS for drafting a beautiful set of figures.

I would like to thank the people working on the IFFN project who helped me in Washington, and who gave the documentation without which this thesis might not have been: Colonel David Archino and Major Michael Gray from the IFFN Joint Test Force, Mr. Earl Hicks from ACS/Studies and Analysis in the Department of the Air Force, and Dr. Mary Ellen Hecker from Logicon, Inc..

I wish to thank all my friends at LIDS for their help and support: Vicky, Muriel, Anne-Claire, Chris, Ashima, Pascal, Herve, Scott, Stamos, and John.

Lastly, I would like to thank my wife Sophie for having been so helpful and supportive, and without whom this thesis might not have been completed under such good conditions.

This research was conducted at the MIT Laboratory for Information and Decision Systems with support provided by the French Delegation Generale pour l'Armement and with partial support provided by the Joint Directors of Laboratories through the Office of Naval Research under contract no. N00014-85-K-0782.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

A major focus of the research in Command Control and Communication (C^3) is the need to assess quantitatively the utility of C^3 systems. Over the past decade, methodologies have been proposed that provide system developers with powerful tools for evaluating the effectiveness of systems [Mishan 1976, Dersin and Levis 1981, White 1985]. All these methodologies assume that experimental data can be gathered for the system to be evaluated.

A new problem arises when very large scale economic, social or military systems are considered because it is often not possible to run a sequence of experiments and collect the data necessary to carry out the assessment. A second feature of these systems is that their development lasts, generally, many years. Indeed, because of budget and time constraints, they are built progressively as components get added gradualy. These large scale systems change with time, and developers seek to optimize their evolution.

A common approach to analyse and evaluate system design is to build a test bed which provides the developer with the ability to consider many different configurations of the same system and with a means to gather data on these configurations in order to evaluate theim. Because of the high degree of complexity of the system, and because of the need for fidelity between actual system behavior and test bed results, the test bed is often very complex too. Indeed, it is a surrogate which is supposed to capture all the properties of interest of the real system. It contains a mix of actual equipment and software and simulations. The construction of a test bed addresses the two problems outlined above: first, it allows one to exercise the system (the test bed in fact) in a reasoned and directed manner; second, it enables one to simulate the evolution of the system by evaluating the effectiveness of different configurations.

Once a test bed has been built, the challenge is to determine whether the actual system is effective and whether and how it can be improved. The developer is interested in evaluating the effectiveness of many different configurations of the system and in selecting an optimal sequence of improvements that will transform gradually the initial system into a

more effective one. For each configuration of the system, one must design experiments to run on the test bed in order to obtain data and assess the effectivenness of this configuration. Then, one must evaluate changes and determine the optimal sequence of configurations (improvements) for the system.

In response to these needs, a methodology is developed in this thesis: the first step is the introduction of a procedure for designing experiments to run on a test bed and evaluating the effectivenness of any configuration of the system; the second step is the development of an algorithm for optimizing the evolution of the system on the basis of the effectiveness evaluation.

This methodology draws on the framework first developed by Dersin and Levis (1981), and then applied to C^3 systems by Bouthonnier (1984), Cothier (1984) and Karam (1985). The method of analysis used is based on relating the performance of a system to the requirements of the mission is has to fulfill.

1.2 OUTLINE OF THE THESIS

This thesis is organized into eight chapters as follows. Chapter 2 is devoted to the detailed presentation of the System Effectiveness Analysis methodology. In this chapter revised definitions are introduced to enable one to draw unambiguously the physical and temporal boundaries of a system.

A procedure for designing experiments to be run on a test bed, and for evaluating the system's effectiveness is presented in Chapter 3.

In Chapter 4, a dynamic programming algorithm is given; this algorithm enables one to determine an optimal sequence of improvements for a system.

Chapter 5 is devoted to the presentation of an air defense system and of its test bed while in Chapter 6 the System Effectiveness Analysis (SEA) is carried out using a mathematical model of the system.

In Chapter 7, the methodology introduced in Chapters 3 and 4 is applied to the air defense system presented in Chapter 5.

Finally, the basic ideas developed in this thesis are summarized in Chapter 8; recommendations for future research are also given.

CHAPTER 2
SYSTEM EFFECTIVENESS ANALYSIS (SEA)

2.1 PURPOSE OF SYSTEM EFFECTIVENESS ANALYSIS

The objective of System Effectiveness Analysis is not to evaluate the performances of a system per se, but rather, to measure the extent to which a system, given its performance, is effective in meeting the mission it is designed to accomplish. One must first define the four basic concepts introduced by the theory: the System, the Mission, the Environment, and the Context.

2.2 DEFINITIONS

Before defining and measuring the effectiveness of a given system, one must first define what is meant by "System". The goal of the following section is to define what the system is for which effectiveness is to be measured.

2.2.1 System, Environment, Context

Let us consider a set U which represents the universe: U will be called the universal set. It may contain a great diversity of elements such as physical entities, data bases, or doctrines. A goal is defined to be a particular desired state of the universe U. Typical goals are : to produce data, to transmit information, or to defend one's assets.

A system S is defined to be a set of elements of the universe U that act together by exchanching information (connectivity) toward the achievement of a particular goal. The set S is a subset of U.

An element u of the universe is included in the environment E if and only if, u does not belong to S, and, the system can act upon u, and, u can act upon the system.

The context C then is defined as the complement of the set S » E in the universe .

With these three definitions one can easily deduce the following properties:

$$U = S \cup E \cup C \quad , \qquad S \cap E = \emptyset$$

$$S \cap C = \emptyset \quad , \quad E \cap C = \emptyset \tag{2.1}$$

where "$\cup$" denotes the union of two sets, and "$\cap$" their intersection.

One can represent these three concepts in a single graph as shown in Fig.2.1.

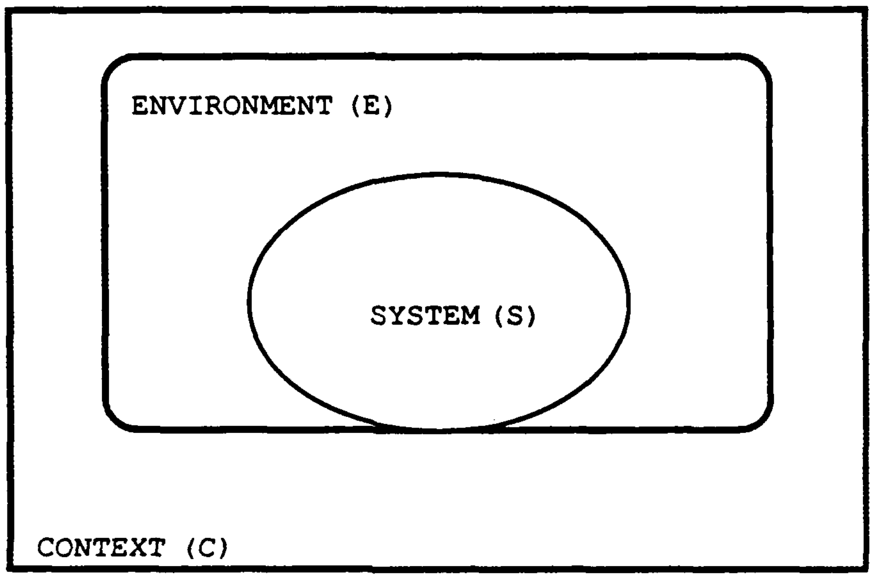


Fig.2.1: System, Environment, and Context

Guidelines

To draw the boundary of S, one must follow three major steps:

(1) Identify the goal the system is designed to achieve

(2) Define the first element of the system; this element is closely related to the goal: indeed, a goal being a particular state of the environment, one can figure out which elements of U act directly on the environment to achieve the goal. One can denote this stage of the system definition, stage S_0.

(3) Define in an incremental approach the next elements of S on the basis of two criteria: connectivity with the components already included in S, and degree to which a given element acts toward the achievement of the goal. Let us suppose that the system definition is at stage S_{n-1}, then one must define stage S_n. To do so, consider all the

components connected to S_{n-1} (connectivity criteria). For each of these components, determine whether it acts toward the same goal as S_{n-1} does; if it does, this component is to be included in S_n. If there are no more elements connected to S_{n-1}, or if each element connected to S_{n-1} does not act toward the same goal as S_{n-1} does,then $S=S_{n-1}$.

Once the system S has been defined, one must determine the environment E. In the complement of the set S, the environment is composed of the elements of the universe that the system can affect and that can affect the system: this definition is simple to apply once the system S has been determined.

2.2.2 Mission

The term "mission" designates the task the system has to perform. It is the particular state of the environment that has to be achieved by the system.

The mission depends on the system considered: for instance, the mission of a C^3 system is to provide "adequate" information to effectors on the basis of the data it receives from the sensors; for a C^3 system, sensors and effectors are part of the environment, according to the definitions sketched out above. The mission of the forces this C^3 system supports is defined in term of the environment of the forces which is different from that of the C^3 system itself.

2.3 TEMPORAL BOUNDARY OF A SYSTEM

2.3.1 Introduction

In the previous section, System, Environment and Context have been defined in order to apply the System Effectiveness Analysis methodology to well defined situations; however, these definitions deal only with static, or non-evolving systems. In this section, these definitions are extended to account for the particuliar features of systems that change as time goes by (evolving systems).

These definitions will enable one to draw the temporal boundary of a system.

2.3.2 System, Environment and Context as fuzzy sets

In section 2.2, System, Environment and Context have been defined as subsets of

the universal set U. One can extend these definitions to fuzzy set ones [Dockery 1985].

Two membership functions , μ_S and μ_E, can be defined for each element of the universe: they represent the fuzzy set extensions of, respectively, the system and the environment.

$$0 \leq \mu_S \leq 1 \ , \ 0 \leq \mu_E \leq 1 \tag{2.2}$$

If $\mu_S(u) = 1$, u is said to be an element of the system.

If $\mu_S(u) = 0$, u is not element of the system.

Otherwise, u belongs only in part to the system; this may be the case for a data-base, a computer, or a communication link shared by two distinct systems.

Similarly, an element u of the universe may be partially included in the environment.

Since an element that is included neither in the system nor in the environment must be included in the context, the membership function of C is defined by:

$$\mu_C = 1 - \mu_S - \mu_E \tag{2.3}$$

Example

For an anti-aircraft battery included in a larger system S, the membership function decreases as missiles get fired, and not replaced, because the contribution of the battery (connected to the large system) to the overall goal of S goes down.

Fuzzy sets will be used in defining evolving systems.

2.3.3 Evolving Systems

A system is defined by its membership function μ_S which may vary over time. If i is an index of the elements in the universe U, let us consider two families of real positive numbers that are inferior to 1 : $(g_{si})_i$ and $(l_{si})_i$.

A system S is said to be "non evolving" during the time interval [t',t"] if and only if, for all elements u_i in the universe and for all time t included in [t',t"] :

$$g_{si} \geq \mu(u_i) \geq l_{si} \tag{2.4}$$

The bounds g_{si} and l_{si} are, respectively, the upper and the lower bound of the membership functions for each element (indexed by i) in the system.

For example, a computer with five processors may be included in a particular system, if at least three of its processors are working; otherwise, the computer is not included in the system because its contribution to the overall mission is assumed to be negligible if two or more processors are out of order. This fact can be modeled by a membership function and a range of variation for this function.

If the universe has only two elements u_1 and u_2, one can define a non-evolving system by a membership function and a set of bounds (g_1,g_2,l_1,l_2) included in $[0,1]^4$.

$$g_{si} \geq \mu(u_i) \geq l_{si} \qquad i=1,2 \tag{2.5}$$

The definition of a non-evolving system can be further extended to a "quasi-static non-evolving system" (Fig.2.2), by allowing the two families $(g_{si})_i$ and $(l_{si})_i$ to vary over time; i.e. the upper and lower bounds for the membership functions are allowed to vary.

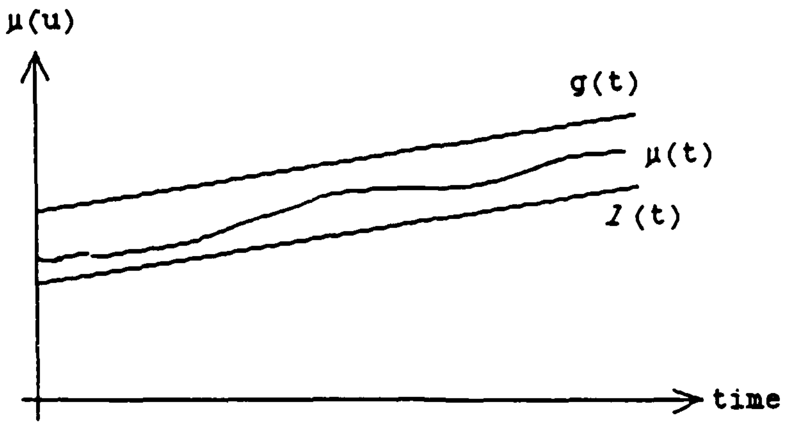


Fig.2.2: Quasi static non-evolving system

In Fig.2.3, the system evolves at time t.

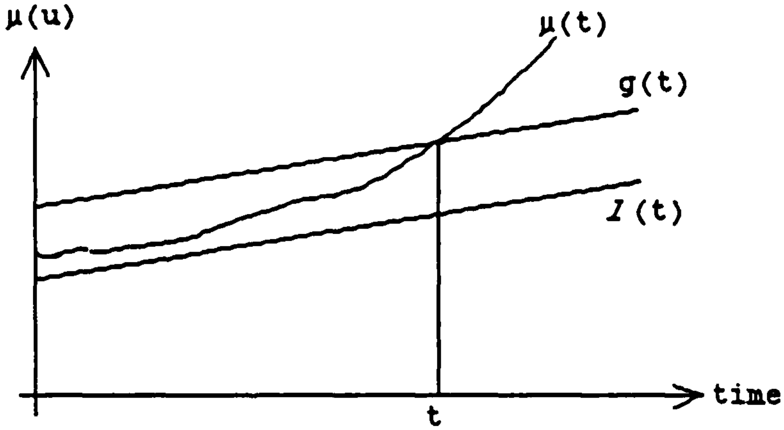


Fig.2.3: Evolving system

2.3.4 Conclusion

The definitions provided in this section are useful for drawing the temporal boundary of a given system: Indeed, the core of the SEA methodology deals with static systems whose temporal boudary must be determined before going into the details of the analysis. The above definitions enable one to determine this boundary in a systematic manner.

2.4 PARAMETERS AND MEASURES OF PERFORMANCE

Parameters are the independent variables in system effectiveness analysis. The system parameters are entities whose value determine system behavior and specify system structure. They are defined within the system boundary. Environment or context parameters refer to the independent variables that describe the environment or the context.

Measures of performance (MOPs) are measurable quantities that describe system properties or attributes. Their values depend on the values of the parameters that characterize the system, the environment and the context.

2.5 SYSTEM LOCUS

The system measures of performance vary in Ω, a subset of R^n where n is the number of MOPs. Since the parameters are constrained to be in a subset P of R^p where p is the number of parameters, one cannot expect the MOPs to take any value in Ω. Since each MOP is a function of several parameters, one can define a mapping from the parameter space into the MOP space (Fig.2.4): This mapping is obtained by exercising the system (or by running simulations) for different contexts and different values of the system parameters (Fig.2.4) in order to generate the reachable MOP values. The set of this reachable MOP values is the system locus L_s.

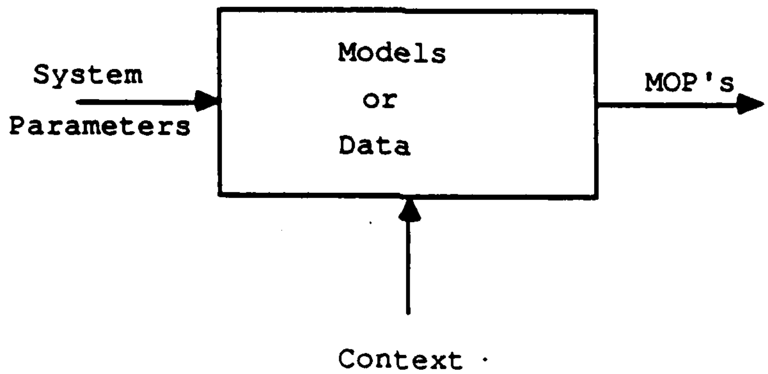


Fig.2.4: Generation of MOP values

This mapping defines the system locus L_s. It is the subset of Ω such that for each $\underline{x}$ in L_s there exists $\underline{p}$ in the parameter space such that $\underline{x}=\underline{x}(\underline{p})$ (Fig.2.5).

2.6 MISSION LOCUS

The mission the system has to achieve is defined by requirements in the MOP space. These requirements are obtained by running models, games or plans for different contexts and for different mission parameters (Fig.2.6).

In order to enable one to compare the mission and the system in the next

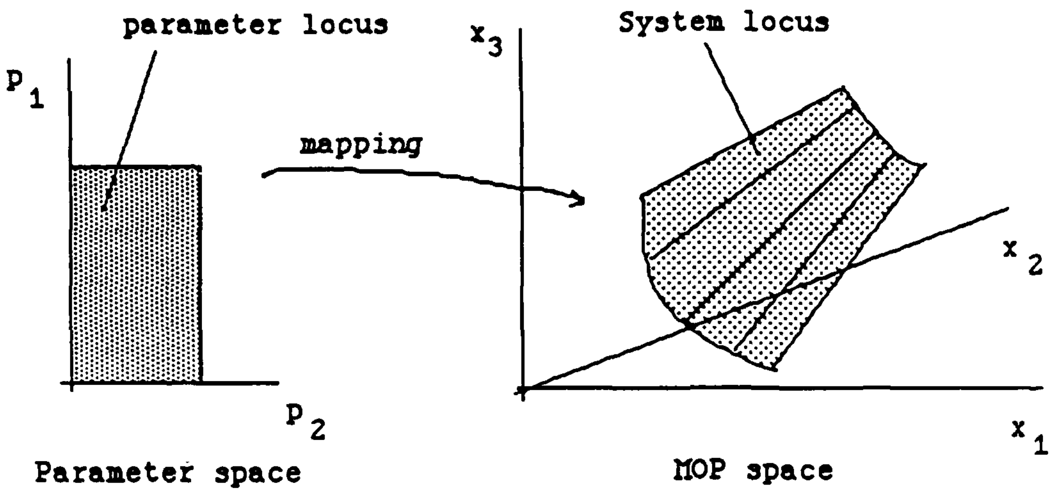


Fig.2.5: System locus

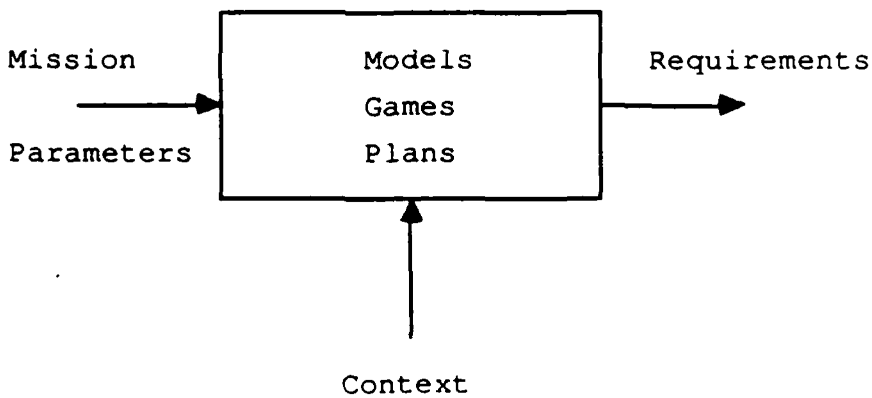


Fig.2.6: Generation of mission requirements

subsection, the requirements obtained by running models in Fig.2.6 must be expressed in term of the MOPs defined for the system: the mission MOPs (expressing the mission requirements) must be the same as the system MOPs (expressing the system capabilities).

The set of MOP values that satisfy the mission requirements constitutes the mission locus L_m (Fig.2.7). The mission locus can be written as follows:

$$a_i \leq g_i(\underline{x}) \leq b_i \quad , i=1,2,...,n \tag{2.7}$$

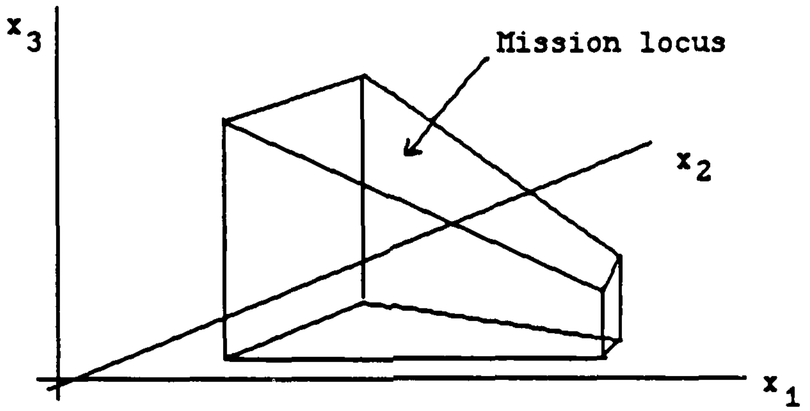


Fig.2.7: Mission locus

2.7 MEASURES OF EFFECTIVENESS (MOEs)

Measures of effectiveness are derived from the comparison of the two loci L_s and L_m . Qualitatively, the greater the intersection of the two loci , the more effective the system is. If V(L) is a measure on the locus L (Fig.2.8), one can define the following MOEs:

$$E_1 = V(L_s \cap L_m)/V(L_s) \tag{2.8}$$
$$E_2 = V(L_s \cap L_m)/V(L_m) \tag{2.9}$$
$$E_3 = V(L_s \cap L_m)/V(L_s \cup L_m) \tag{2.10}$$

E_1 is the degree to which the system capabilities are included in the mission locus; It measures how well the system capabilities are used for the mission considered.

E_2 is the degree to which the mission locus is covered by the system; It is the

degree of coverage of the mission by the system.

E_3 is the degree of overlapping between the system capabilities and the mission requirements; $1-E_3$ is the degree of misfit between the two loci.

If $V(L_s \cap L_m)$ is not equal to zero, then one can deduce the following relation between the three MOEs defined above:

$$1/E_1 + 1/E_2 = 1 + 1/E_3 \tag{2.11}$$

The important fact in the passage from an MOP to an MOE is the addition of requirements: Before setting requirements, the system locus doesn't tell how effective the system is.

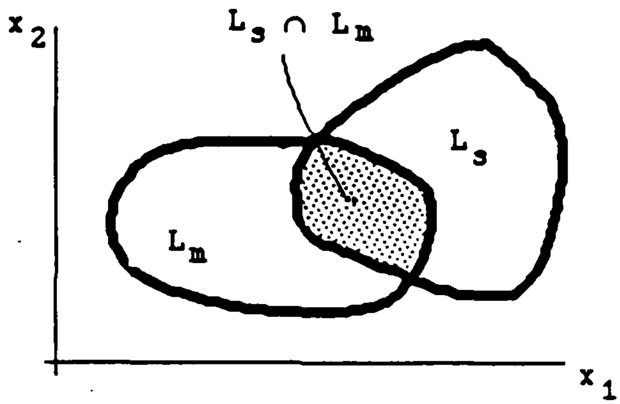


Fig.2.8: Measures of effectiveness

Special cases: If L_s is included in L_m, then $E_1=1$ and $E_2=E_3<1$. If L_m is included in L_s, then $E_2=1$ and $E_1=E_3 < 1$.

For a given system one can define many different MOEs representing effectivenesses from various standpoints: These MOEs are called partial MOEs. Let $E_1, E_2, \ldots, E_p$ be the partial MOEs for a given system. Debreu [1968] has shown that, under certain conditions, there exists a real valued function, a utility function, which is continuously dependent on the E_i. Let U be such function taking values between 0 and 1:

$$[0,1]^p \longrightarrow [0,1]$$
$$E_1, E_2, \ldots, E_p \longrightarrow U(E_1, E_2, \ldots, E_p)$$

Then, the global measure of effectiveness can be taken to be $U(E_1, E_2, ..., E_p)$.

2.8 STEPS OF THE METHODOLOGY

The steps of the System Effectiveness Analysis and their interrelationships are shown in Figure 2.9 and are summarized below (Step 1 to 8 are executed for each subsystem).

Step 1: Define the System, the Environmnent the Context and the Mission.

Step 2: Define from the system characteristics, the Environment, and the Context, the independent variables (parameters) that are l ikely to influence the System MOPs.

Step 3: Relate the System MOPs to the parameters.

Step 4: Define from the mission characteristics, the Environment, and the Context, the independent variables (parameters) that are likely to influence the Mission MOPs. In order to enable one to compare the mission and the system, the mission MOPs must be the same as the system MOPs.

Step 5: Relate the Mission MOPs to the parameters.

Step 6: Compute the System locus.

Step 7: Compute the Mission locus (or mission requirements).

Step 8: Compute the partial measure of effectiveness.

Step 9: Combine the partial measures of effectiveness obtained for each subsystem into a global measure of effectiveness for the system as a whole.

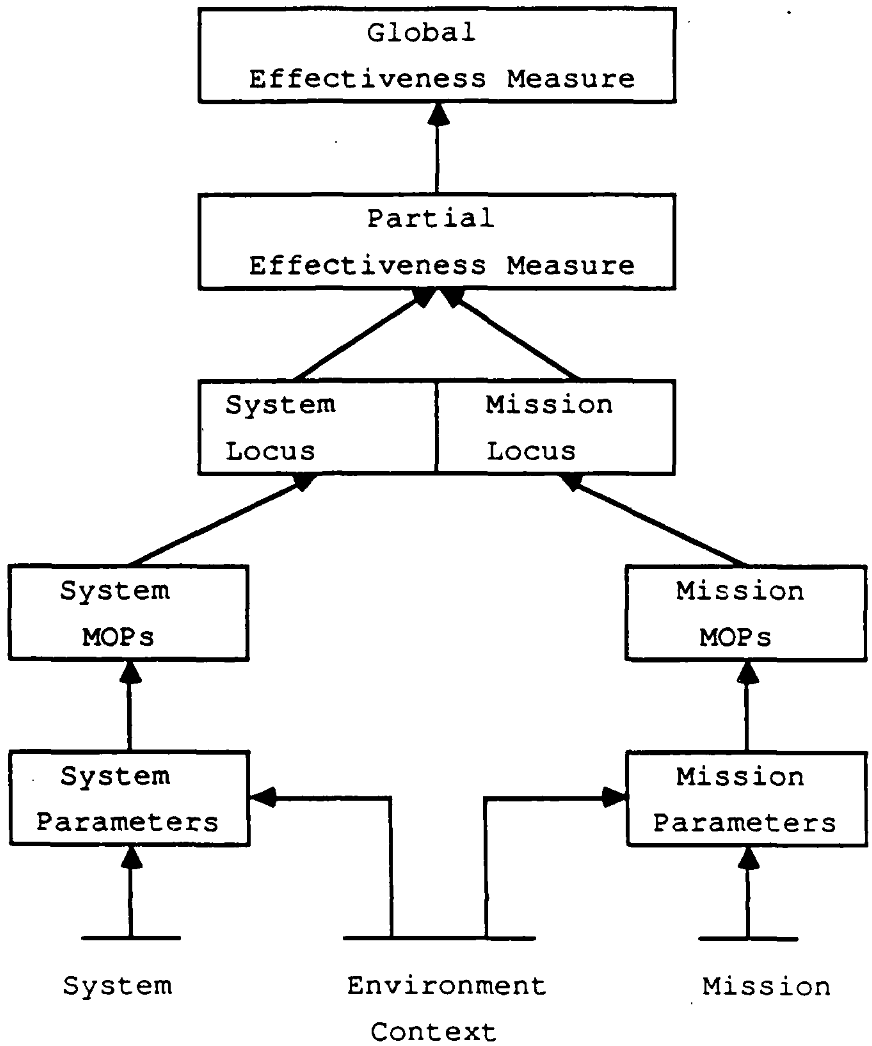


Fig.2.9: System Effectiveness Analysis Methodology

2.9 CONCLUSION

The framework of the System Effectiveness Analysis has been presented throughout chapter 2; this methodology constitutes the basis of the analysis to be carried out in this thesis in order to design experiments to run on large scale systems and to improve these systems.

Chapter 3 will present a experiment design algorithm to evaluate the effectiveness of real systems as opposed to any model of these systems.

CHAPTER 3
EXPERIMENT DESIGN FOR LARGE SCALE SYSTEMS

3.1 INTRODUCTION

In this chapter, a procedure for designing experiments to run on a large scale system is developed. This procedure will enable one to assess the effectiveness of an actual system and, eventually, improve it.

We assume that we have a large scale system that cannot be exercised as one wants and whose effectiveness has to be assessed: in order to do so, one needs to gather data concerning this large scale system; indeed, to assess the effectiveness with the SEA methodology, one must determine the system locus of the system at hand. This requires operating the system at an extremely large number of sets of parameter values.

A common approach to gather data on such a system is to build a test-bed which is assumed to be a complete surrogate of the system: this test-bed can be exercised as one wants, but, since it is often expensive and time consumming to operate, only a small number of experiments can be run. The procedure presented in this chapter provides a means to determine a small set of experiments to be run on the test-bed: the small number of experimental values will be combined with the results obtained from a simplified mathematical model of the system to construct the system locus of the test-bed. The underlying assumption is that the test-bed is an excellent approximation of the actual system. Because of this assumption, no distinction will be made in this chapter between the actual system and its surrogate, the test-bed; the only distinction that will be considered is the one between the simplified mathematical model and the actual system (or the test-bed). After determining the system locus of the actual system, one can evaluate its effectiveness.

3.2 OVERVIEW OF THE PROCEDURE

As mentioned in the introduction, the effectiveness of a large scale system cannot be evaluated directly; indeed, since the actual system cannot be exercised as one wants, it is

impossible to determine its system locus: it would require to run an extremely high number of experiments.

In this chapter, a simplified mathematical model of the actual system is considered; this model can be represented by a mapping "f" from the parameter space into the MOP space. Two mappings, and consequentely two loci can be considered. First, using the mathematical model "f", the parameter space can be mapped into the "model" system locus. Second, the actual system, if it could be exercised for all the values of the parameter space, would yield the "actual" system locus (Fig.3.1). Since the model is a simplified one, the model locus is an approximation of the actual locus.

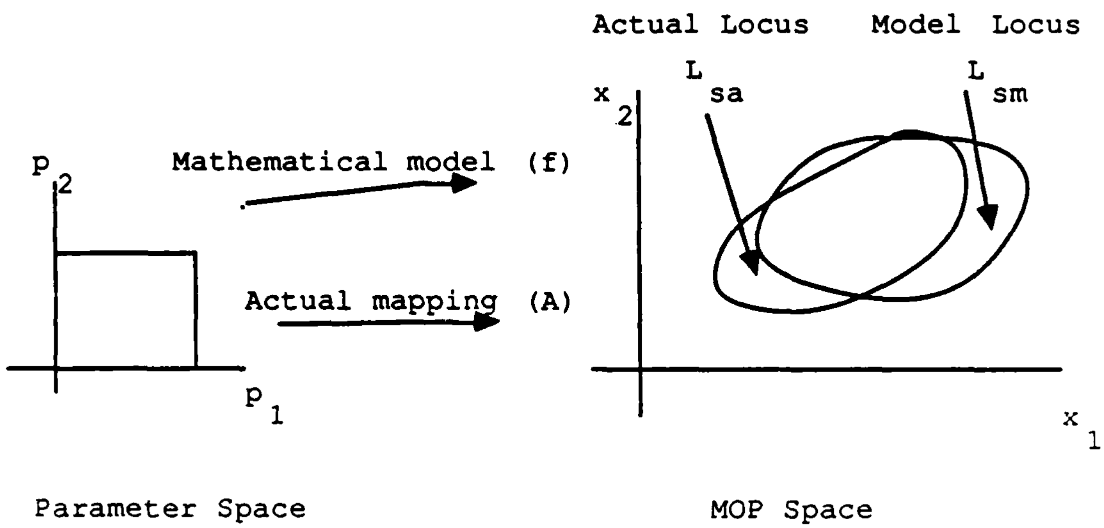


Fig.3.1: Actual and model loci

This chapter aims at determining the actual locus of the system; the key idea is to obtain the model locus, and a few points of the actual locus, and to determine a mapping "T" that transforms the model locus into the actual locus in the MOP space. This mapping T that goes from the MOP space to the MOP space will be obtained from the model locus and the few points determined through experimentation of the actual locus. Then, the actual locus will be obtained indirectly: if "A" is the actual mapping from the parameter space into the MOP space, then A=Tof, where "o" denotes the composition of two functions (Fig.3.2). With this algorithm, only a few points belonging to the actual locus, and therefore only a few experiments will be necessary to evaluate the locus of the actual

system and eventually the effectiveness of this system.

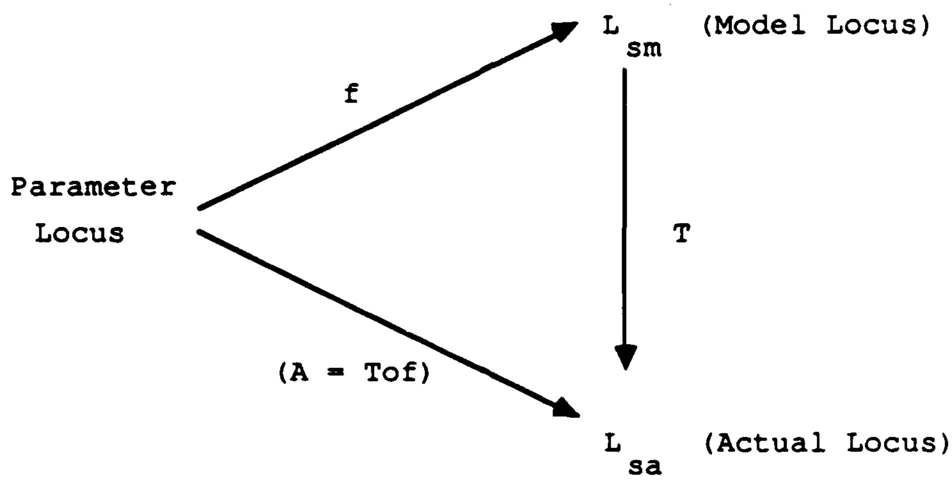


Fig.3.2: Determination of the actual mapping

On Fig.3.2 the actual mapping A is denoted (A = Tof) to emphasize the fact that it cannot be obtained directly but only as a composition of f and T.

Since f is assumed to be known, only T has to be evaluated if one wants to determine the actual mapping. The focus of the remaining part of this chapter is therefore the determination of T.

The first step is to obtain a small number of points in the actual locus. In sections 3.3 and 3.4, tools for selecting these points are introduced: they will be used in section 3.5 where the actual locus will be constructed.

In the next section, an inversion algorithm that will be used later in this chapter for selecting points in the actual locus is presented.

3.3 INVERSION ALGORITHM

3.3.1 Introduction

In this section an algorithm is presented that allows one to determine a parameter vector corresponding to a given MOP vector in the model locus. Later in the development, this alghorithm will be applied to a small number of points in the model locus to obtain an

equal number of parameter vectors. Experiments will be run at these parameter values to yield the small number of points belonging to the actual locus we are looking for.

Let us suppose that we want to reach a desired point in the MOP space; the purpose of the inversion algorithm presented below is to find a value in the parameter space that yields through "f" the desired value in the MOP space: it is to invert the mathematical model "f" for a given point in the MOP space. The mapping "f" is not assumed to be invertible for each MOP value; in addition, the dimension of the parameter space and of the MOP space may be different: therefore, the inversion algorithm must determine whether "f" can be inverted at a specific MOP value, and if it can, the algorithm will yield a parameter vector (which may not be unique) corresponding to the desired MOP value.

3.3.2 Notation

A system with m MOPs $\{x_i | i=1,...,m\}$ and n parameters $\{p_j | j=1,...,n\}$ is considered; Generally, n and m are not equal. If "f" is the mapping from the parameter space into the MOP space, then $\underline{x}=f(\underline{p})$ where $\underline{x}$ and $\underline{p}$ are column vectors. If $\underline{p}_0$ is an initial combination of parameters, the corresponding point in the MOP space is $\underline{x}_0=f(\underline{p}_0)$; $\underline{p}_0$ will be called a "basic operating point".

3.3.3 The inverse problem

The next objective is to determine the small variation $\underline{dp}$ around $\underline{p}_0$ coresponding to a small variation $\underline{dx}$ around $\underline{x}_0$. Since f is generally non-invertible, one has to find an algorithm that determines $\underline{dp}$. If f is assumed to be differentiable at $\underline{p} = \underline{p}_0$, then

$$\underline{x} = \underline{x}_0 + [df(\underline{p}_0)/dp].\underline{dp}$$

If $\underline{G} = [df(\underline{p}_0)/dp]$ then,

$$\underline{G} = (g_{ij}) \text{ with } g_{ij}=[\partial x_i(\underline{p}_0)/\partial p_j],\ i=1,...,m\ ;\ j=1,...,n$$

where $\underline{G}$ is a m by n matrix.

<u>Singular value decomposition of G</u>

The singular value decomposition of $\underline{G}$ yields:

$$\underline{G}=\underline{U}.\underline{\Sigma}.\underline{V}' \quad \text{and} \quad \underline{\Sigma}=\underline{U}'.\underline{G}.\underline{V},$$

where $\underline{U}$ is a m by m unitary matrix, $\underline{V}$ a n by n unitary matrix, and $\underline{\Sigma}=(s_{ij})$ an m by n

matrix with $s_{ij}=0$ if $i \neq j$ and $s_{ii}=s_i \geq 0$. The s_i are the singular values of $\underline{G}$. If $(\underline{U}_i)$ are the column vectors of matrix $\underline{U}$, and $(\underline{V}_j)$ the column vectors of matrix $\underline{V}$, then

$$\underline{G}.\underline{V}_i=s_i.\underline{U}_i \, .$$

Moreover, $(\underline{V}_i),i=1,...,n$ is a basis of R^n and $(\underline{U}_j),j=1,...,m$, is a basis of R^m.

Application

If one considers a small variation Δ in the MOP space,one can decompose Δ as follows:

$$\Delta=D_1\underline{U}_1+D_2\underline{U}_2+....+D_m\underline{U}_m \, .$$

If $s_i \neq 0$ for $i=1,...,m$ then a corresponding direction in the parameter space is

$$\underline{d}=(D_1/s_1)\underline{V}_1+(D_2/s_2)\underline{V}_2+...+(D_m/s_m)\underline{V}$$

This direction may not be unique if $n>m$: Indeed, if $n > m$, $\underline{d}+\underline{V}_i$ $n \geq i > m$ satisfies the problem, because $\underline{G}.\underline{V}_i=0$ for $n \geq i > m$.

If some singular values are equal to zero, it means that there are points unreachable from $\underline{x}_0$.

Therefore if s_i, $i=1,...,m$, are the singular values of A and if $s_i \neq 0$ for $i=1...k$ $(k<m)$ and $s_i=0$ for $i=k+1...m$ then:

•For any direction in the MOP space such that

$$\Delta=D_1\underline{U}_1+D_2\underline{U}_2+...+D_k\underline{U}_k \qquad (3.1)$$

one can find a corresponding direction in the parameter space:

$$\underline{d}=(D_1/s_1)\underline{V}_1+(D_2/s_2)\underline{V}_2+...+(D_k/s_k)\underline{V}_k. \qquad (3.2)$$

(Since s_i for $i=1...k$ is not equal to zero, one can divide by s_i).

•For any direction

$$\Delta=D_{k+1}\underline{U}_{k+1}+D_{k+2}\underline{U}_{k+2}+....+D_m\underline{U}_m$$

with one of the D_i (i=k+1,....,m) not equal to zero, it is impossible to find a corresponding direction in the parameter space.(Indeed, if for example $s_{k+1}=0$, it means that U_{k+1} is not element of the image space of the paramater space.)

Therefore, the singular value decomposition of the matrix $\underline{G}$ provides a means to find a small variation in the parameter space that corresponds to the desired variation in the MOP space. It may be impossible to find such a direction if the desired direction in the MOP space is not element of the image space of the parameter space through $\underline{G}$; the algorithm tells us whether it is possible or not. In general, when it is possible to find a corresponding direction in the parameter space, this direction is not unique; in this case, there are some degrees of freedom in choosing this direction $\underline{d}$. For example, one can use the smallest norm of the vector obtained in the algorithm, or the smallest number of parameters affected in the parameter space, as rules for selecting the direction $\underline{d}$. In what follows, we will always choose the direction determined by the vector $\underline{d}$ whose euclidian norm

$$\|\underline{d}\| = [(\underline{d}_1)^2 + (\underline{d}_2)^2 + \dots + (\underline{d}_m)^2]^{1/2} \tag{3.3}$$

is the smallest: indeed, since we are dealing with a linearized function, a large norm $\underline{d}$ will degrade the accuracy of the algorithm if the gradient matrix $\underline{A}$ is ill-conditioned; another reason for choosing this direction is that we want the new parameter value to be as close as possible to the initial one: it corresponds to the slightest variations of the operating conditions of the system.

3.3.4 Single stage parameter value determination

With the notation defined above, it is possible to explore the system locus around $\underline{x}_0$; If for instance, one wants to reach point $\underline{x}_d$ (for $\underline{x}$-desired) close to $\underline{x}_0$ in the MOP space, one can apply the procedure described above to $\underline{\Delta}=\underline{x}_d-\underline{x}_0$. The outcome is a vector $\underline{d}=\underline{p}'-\underline{p}_0$ in the parameter space; Therefore, the parameter value corresponding to $\underline{x}_d$ is $\underline{p}'=\underline{d}+\underline{p}_0$. As stressed above, some points close to $\underline{x}_0$ may be unreachable because of singular values equal to zero. Locally, around $\underline{x}_0$, it is therefore possible to explore the system locus of the model in a directed manner. An interesting point is that to go from $\underline{x}_0$ to $\underline{x}_d$ only m directions are involved in the parameter space.

If one wants to reach $\underline{x}_d$ close to $\underline{x}_0$ in the MOP space, the new operating point $\underline{p}'$ can be computed with the mathematical model and the procedure described above. Since it

is not guaranteed that the $\underline{p}'$ computed in the singular value algorithm will belong to the parameter locus (Fig.3.3), one must consider the projection of $\underline{p}'$ on the parameter locus. In what follows $\underline{p}=\text{proj}(\underline{p}')$ refers to the projection of $\underline{p}'$ on the parameter locus.

Because of the projection $\underline{p}$ of $\underline{p}'$ on the parameter locus, and because of the linearized model used in the inversion algorithm, one cannot expect that the simplified model f applied to the new parameter value $\underline{p}$ will yield the desired point in the MOP space (Fig.3.3); in general:

$f(\underline{p}) = \underline{x}' \neq \underline{x}_d$ (desired value in the MOP space) (3.4)

because of linearization and because of the projection of $\underline{p}'$ on the parameter locus.

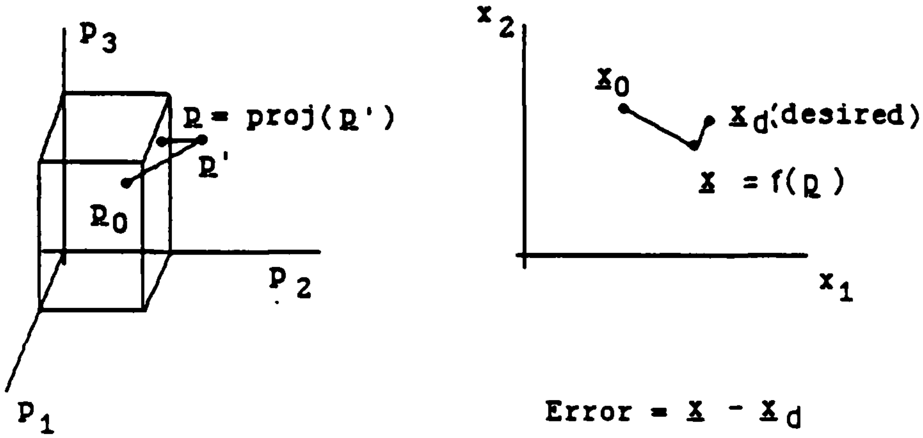


Fig.3.3: Determination of $\underline{p}$, and error in the MOP space

3.3.5 Multi-stage parameter determination

As we have just seen, the single step parameter determination generally yields an inaccurate approximation in the MOP space. If one is interested in an accurate determination, one can go through the inversion algorithm again and again, until the error vector is small enough for the application at hand. For the n^{th} stage of the process, the initial parameter value is the one that has been computed at stage n-1; the mapping f is linearized at that point and the single stage inversion algorithm is applied.

The process stops when the MOP point obtained by applying f to the computed

parameter value is close enough (for the application considered) to the desired point in the MOP space. This multi-stage process is represented as a flowchart in Fig.3.4.

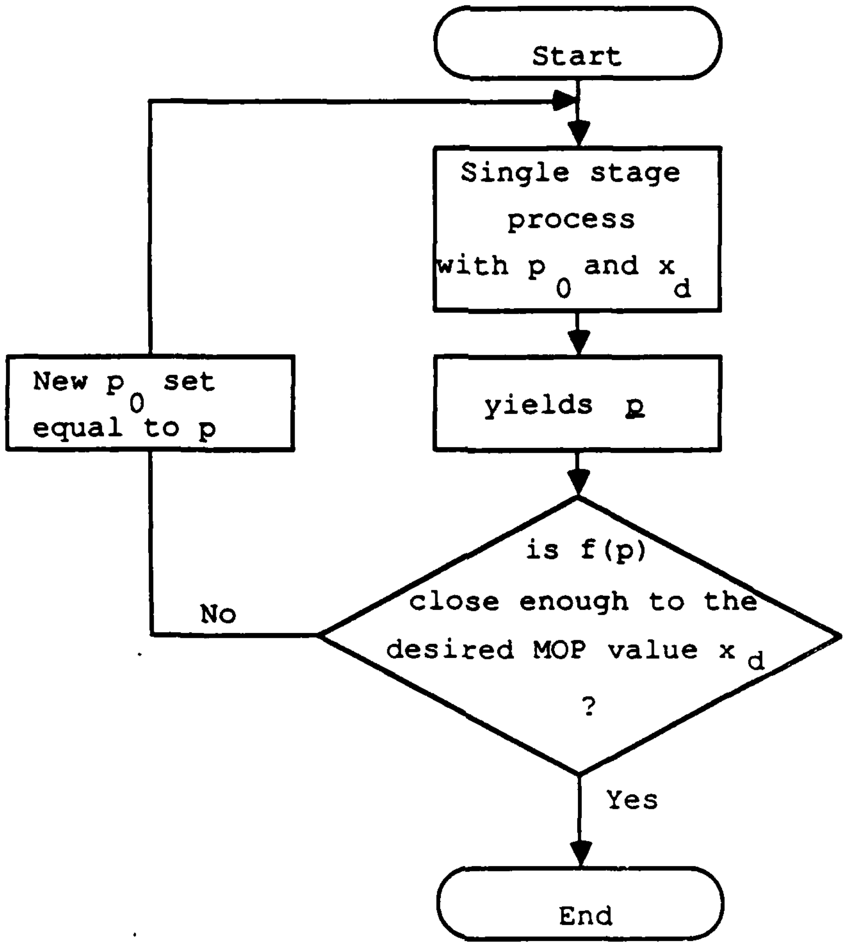


Fig.3.4: Multi-stage parameter determination

Therefore, this inversion algorithm provides a means to find a parameter value such that the simplified mathematical model "f" when applied to this parameter value will yield a point in the MOP space arbitrary close to a desired MOP value.

3.4 EXPERIMENTAL DESIGN

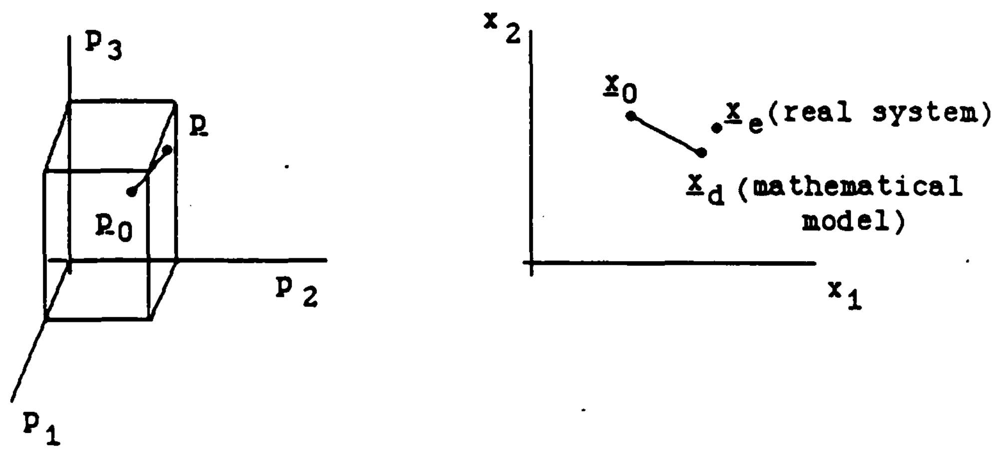
3.4.1 Introduction

After applying the inversion algorithm, we have determined $\underline{p}$ such that: $\underline{x}_d=f(\underline{p})$; the error in the determination of $\underline{p}$ is assumed to be so small as to be negligible. At the basic operating point considered we have $\underline{x}_0=f(\underline{p}_0)$.

One can assume that the simplified model has been calibrated at the basic operating point: a reasonable asssumption is that the simplified model and the actual system coincide at $\underline{p}=\underline{p}_0$:

$$\underline{x}_0 = f(\underline{p}_0) = \text{Actual System } (\underline{p}_0).$$

To determine what the actual system locus looks like, we will choose a small number of desired points in the MOP space ($\underline{x}_d$); how these points are chosen is described in section 3.4.2. For each of these point, with the algorithm presented in section 3, we will find a parameter vector such that $\underline{x}_d = f(\underline{p})$. Then, an experiment will be run on the actual system at this parameter vector: the outcome of the experiment will be a point $\underline{x}_e$ in the MOP space. Since the simplified model and the actual system are close but not equal, the values $\underline{x}_e$ and $\underline{x}_d$ are going to be different (Fig.3.5).



$$\text{desired } \underline{x}_d \xrightarrow[\text{of "f"}]{\text{inversion}} \underline{p} \xrightarrow{\text{experiment}} \underline{x}_e$$

Fig.3.5: Mathematical and Experimental MOP values

The focus of this section is the determination of the points $\underline{x}_d$ in the model locus; for each of the points selected in this section, we will apply the inversion algorithm described above; it will yield as many parameter vectors as points $\underline{x}_d$; for each parameter vector computed that way, the corresponding experiment will be run and yield an experimental MOP vector $\underline{x}_e$ belonging to the actual locus. With this procedure, for each desired point selected in the model locus, there will be a corresponding point in the actual locus; to a given choice of points in the model locus will correspond a selection of points in the actual locus.

3.4.2 Selection of points in the MOP space

The first step that has to be completed in the experimental design process presented above is the selection of desired values in the MOP space: one must select a small number of vectors $\underline{x}_d$. These vectors $\underline{x}_d$ will be denoted as an indexed family $(\underline{x}_{di})$ i=1,...,r. The number of points selected will be exactly the number of experiments that will be run on the actual system.

The only available information for selecting the points $\underline{x}_d$ is the system locus obtained from the mathematical model. With the mapping f, one can easily determine this locus. Since one is interested in having a selection of points that represents the entire locus as opposed to any of its region, the points that will be chosen must be distributed all over the locus. A simple way to choose a small number of points that are representative of the locus is to inscribe it in an n-dimensional rectangular parallilepiped, and to choose the tangeancy points; the two dimensional case is shown in Fig.3.6. This method for choosing the points in the model locus is not only simple; it is the one that allows one to select the minimum number of points for looking at the whole locus.

The vectors $\underline{x}_d$ that we are going to consider are the points where the system locus and the parallilepiped determined above are tangent (Fig.3.6). We will denote r the number of these vectors $\underline{x}_d$; in each dimension, they correspond to the maxima and minima of each MOP.

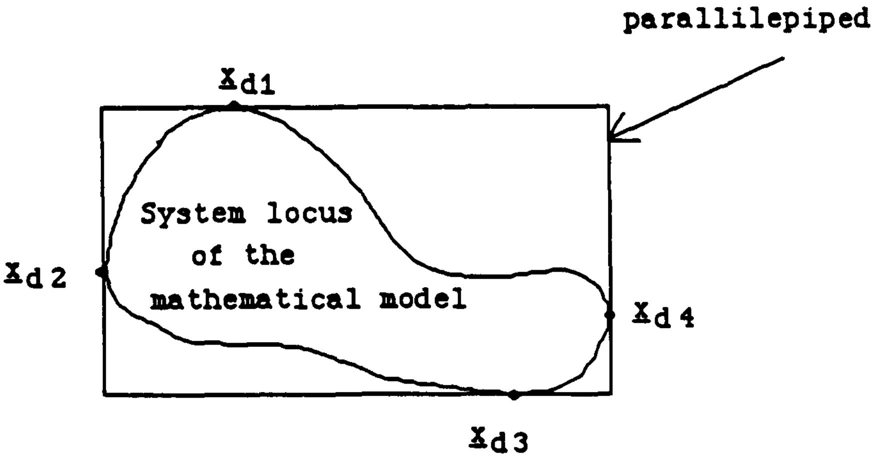


Fig.3.6: Determination of the x_d

In general, more than one point in the system locus will be tangent to the parrallilepiped defined above: for example, the system locus itself may be a parallilepiped, or part of the system locus boundary may be a plane surface included in one of the surface that defines the parallilepiped (Fig.3.7); there will be an infinite number of tangency points in these two latter situations. The number of tangency points may also be finite (Fig.3.7). Since we want to choose only a small number of points, and since we are interested ultimately in determining how the parallilepiped is deformed when one goes from the mathematical locus to the actual one, we will consider only one point of tangency for each surface defining the parallilepiped. This point will be the center of gravity of all the tangency points for each surface of the parallilepiped; if the number of points is finite, the center of gravity may not belong to the locus (the selected points must belong to the model locus if one wants to be able to find a corresponding parameter value): in that case, the closest point of the locus to the center of gravity will be chosen. Given the constraint that one must look at the whole locus, the procedure presented above selects the minimum number of points; it will eventually corresponds to the minimum number of experiments that needs to be run.

Therefore, we will select 2n (i.e. r=2n) points on the mathematical locus, where n is the dimension of the MOP space.

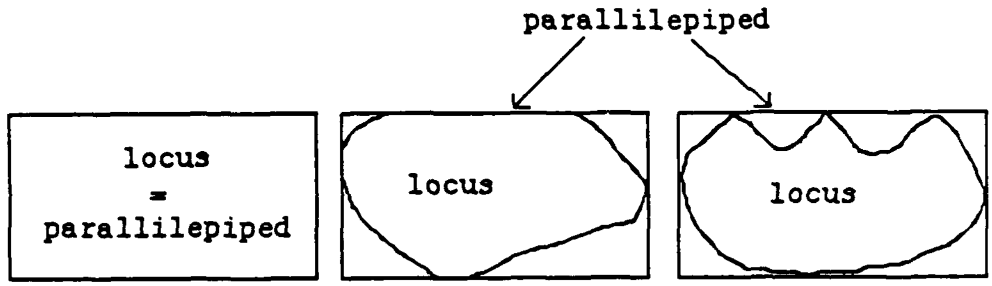


Fig.3.7: Three system locus configurations

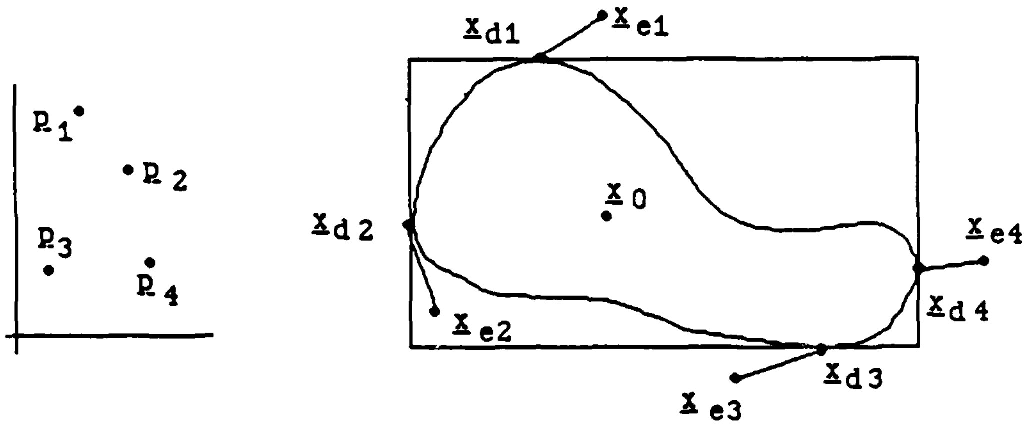
3.5 RECONSTITUTION OF THE ACTUAL SYSTEM LOCUS

3.5.1 Introduction

We now have all the tools required to select a small number of points in the actual locus, and to compute the mapping T that transforms the model locus into the actual locus. First, let us determine the small number of points we are looking for in the actual locus.

3.5.2 Points in the actual locus

We apply the inversion algorithm presented in section 3.3 for the r points ($\underline{x}_{di}$) i=1,...,r selected in section 3.4; it yields r vectors ($\underline{p}_i$) i=1,...,r in the parameter space. Then, experiments corresponding to these parameter vectors are run on the actual system: that is, the experimental conditions are set as required by the parameter vectors. This is always feasible because the parameter vectors determined with the inversion algorithm are constrained to belong to the set of admissible parameters (parameter locus): indeed, they are projected on the parameter locus. As shown in section 3.4, the procedure for choosing the points $\underline{x}_{di}$ is the one that allows one to look at the whole locus with the minimun number of points: thus, given the contraint that one must look at the whole locus, the minimum number of experiments are actualy run. The outcome of these experiments will be r experimental values ($\underline{x}_{ei}$) i=1,...,r in the MOP space (Fig.3.8).



$$\underline{x}_{di} = f(\underline{p}_i) \quad , \quad \underline{x}_{ei} = \text{Experiment}(\underline{p}_i) \quad , i=1..4$$

Fig.3.8: Experimental results

To the r experimental values, one must add $\underline{x}_0$ because an assumption of the procedure is that the simplified model and the actual system coincide at $\underline{x}_0$. In general, we have therefore r=2n+1 experimental vectors with n coordinate for each. On the basis of these results we must determine the system locus of the actual system.

Therefore, with this procedure, a small number of experiments have been run to yield a few (2n+1) points in the actual locus. On the basis of this small number of points and the model f, we will determine the transformation T introduced in section 3.2 that maps the model locus into the actual locus.

3.5.3 From the model to the actual locus

We have only 2n+1 points in the actual locus, where n is the dimension of the MOP space. The focus of this section is to determine, for each single point $\underline{x}_m$ in the model locus, the corresponding point $\underline{x}_a$ in the actual locus: it is the computation of a transformation T that maps the model locus into the actual locus. For each $\underline{x}_m$ in the model locus, we will have $\underline{x}_a = T(\underline{x}_m)$.

We assume that the transformation from the model locus to the actual one is the composition of a translation $\underline{V}$ and a linear transformation L; thus, for any vector $\underline{x}_m$ in the mathematical locus, the corresponding point in the actual locus will be $\underline{x}_a$ with

$$\underline{x}_a = L\underline{x}_m + \underline{V} . \qquad (3.5)$$

Such a transformation is described by n^2+n coefficients in an n-dimensional space (square matrix L with n rows and n columns and vector $\underline{V}$). To determine this transformation, we have rn equations where r is the number of experiments: since for r=2n+1,

$$rn = 2n^2 + n > n^2 + n ,$$

one can implement a least square procedure to compute L and $\underline{V}$.

$$\text{Let } e_i = \| L\underline{x}_{di} + \underline{V} - \underline{x}_{ei} \| \tag{3.6}$$

where $\| \ \|$ represents the euclidian norm (see equation 3.3). The number e_i is the norm of the difference between the actual locus vector obtained by the transformation and the experimental value. Then the transformation (3.4) is the one that minimizes the following expression (r is the number of experiments).

$$J = e_1 + e_2 + \ \dots \ + e_r \tag{3.7}$$

With this transformation T, for each point $\underline{x}_m$ in the model locus, the corresponding point in the actual locus is

$$\underline{x}_a = T(\underline{x}_m). \tag{3.8}$$

This "T" is the same as the one introduced in section 3.2 (Fig.3.2): therefore, we can interpret A as the transformation Tof that maps the parameter locus into the actual system locus.

3.5.4 Conclusion

In this section the tools introduced in sections 3.3 and 3.4 have been used to determine the mapping R=Tof from the parameter locus into the actual locus. This mapping is the outcome of a mathematical model combined with the small number of experimental values that one can usually afford to run on the actual system. It should be noted at that point that, if the actual system could be exercised for all the values of the parameter locus, it would probably yield an actual system locus slightly different from the one obtained at the end of this section; given the experimental constraints, the actual locus obtained with

A=Tof is the best approximation of the actual system locus that would be obtained by running as many experiments as one would like.

3.6 STEPS OF THE PROCEDURE

The steps of the experiment design procedure and their interrelationships are shown in Fig.3.9 and are summarized below:

Step 1: Define a mathematical model "f" of the system at hand.

Step 2: Analyze this model with the System Effectiveness Analysis methodology in order to obtain a mathematical system locus.

Step 3: Determine r points of interest $\underline{x}_{di}$ in the mathematical locus (tangency points with a n-dimensional parallilepiped).

Step 4: With the inversion algorithm, for each $\underline{x}_{di}$, find a parameter vector $\underline{p}_{di}$ in the parameter locus such that $\underline{x}_{di}=f(\underline{p}_{di})$.

Step 5: For each $\underline{p}_{di}$ run an experiment on the actual system: it yields r experimental values $\underline{x}_{ei}$.

Step 6: To the r mathematical MOP points ($\underline{x}_{di}$) and their corresponding experimental points ($\underline{x}_{ei}$), apply the least square procedure to find a transformation T that maps the mathematical locus into the actual locus.

Step 7: Apply A = Tof to each point in the parameter locus to obtain the actual system locus.

Then, with the actual system locus, one can evaluate the effectiveness of the actualsystem, as opposed to evaluating the effectiveness of a simplified model.

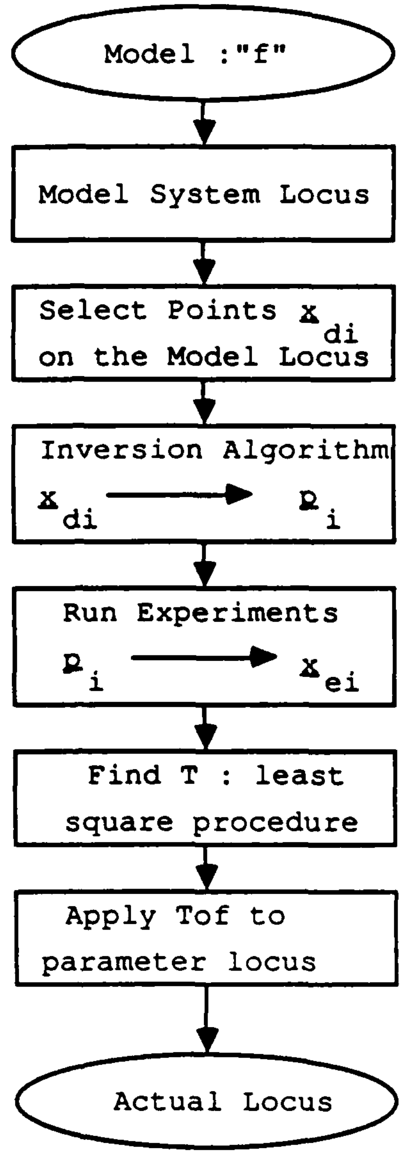


Fig.3.9: Steps of the experiment design procedure

3.7 CONCLUSION

In this chapter, an algorithm to design experiments, and to evaluate the system locus of a actual system on the basis of a simplified model, has been presented. As shown in the development of the chapter the implementation of the selected experiments is always possible. Moreover, by inscribing the model locus into a parallilepiped, the minimum number of experiments that needs to be run has been obtained: this is a deliberate choice

that allows one to look at the whole locus by running as few experiments as possible. Such an algorithm combined with the System Effectiveness Analysis methodology enables one to determine the effectiveness of an actual system as opposed to making the evaluation on a model of this system.

After having evaluated the effectiveness of an actual system, one may be interested in improving it; the purpose of the following chapter is the presentation of a dynamic programming algorithm designed to improve a system over time.

CHAPTER 4
SYSTEM IMPROVEMENT AND DYNAMIC PROGRAMMING

4.1 INTRODUCTION

Since the development of a large scale system lasts generally many years, the developers need tools to plan the evolution of the system at hand; because of budget and time constraints the system is built progressively, and components are added gradualy: therefore, the future modifications of a system are known by the system developer for a given time horizon. During its lifetime, the system will be modified at given time intervals (generally one year intervals for administrative reasons): therefore, the number of stages the system is going to go through can be estimated accurately. During the evolution process, the goal pursued by the system developer is the maximization of the effectiveness over the system lifetime.

In order to carry out this maximization, the sytem developer must determine the optimal sequence of modifications. The new system will be represented in the MOP space by a new locus, and will correspond to a given number of modifications: add or remove such and such components, change a particular structure, and so on. Since all the changes cannot be carried out at the same time, the system will evolve, in the sense defined in chapter 2, as components get added or removed. The problem is to optimize the multi-stage decision process regarding the implementation of these modifications. In this chapter, we are going to present an algorithm that enables one to determine under certain assumptions an optimal sequence of modifications to improve an existing system over time. Since one is interested in optimizing the effectiveness of the actual system as it evolves, the algorithm presented in chapter 3 can be used to determine the effectiveness of the actual system on the basis of a simplified mathematical model at each stage.

Because of the features of an evolution process such as the one outlined above, the basic assumptions underlying this chapter are as follows: first of all, we assume that the possible modifications of the system are known and are constrained to be in a given set; moreover, the number of modifications that the initial system is going to go through is assumed to be known. The second assumption is that the goal of the system developer as

time goes by is the maximization of an objective function which is a linear combination of the MOEs of the system at different stages of its evolution.

4.2 MODELING THE IMPROVEMENT PROCESS

4.2.1 Single stage evolution

Before going into a sequence of modifications, let us model a single stage evolution for a system (Fig.4.1).

Let X_0 be a given system which does not evolve between time t_0 and t_1; MOE_0 is the effectiveness of X_0. If at time t_1 system X_0 evolves because of an exogenous modification U_0, the new system is denoted X_1. A modification U_0 is a decision variable.

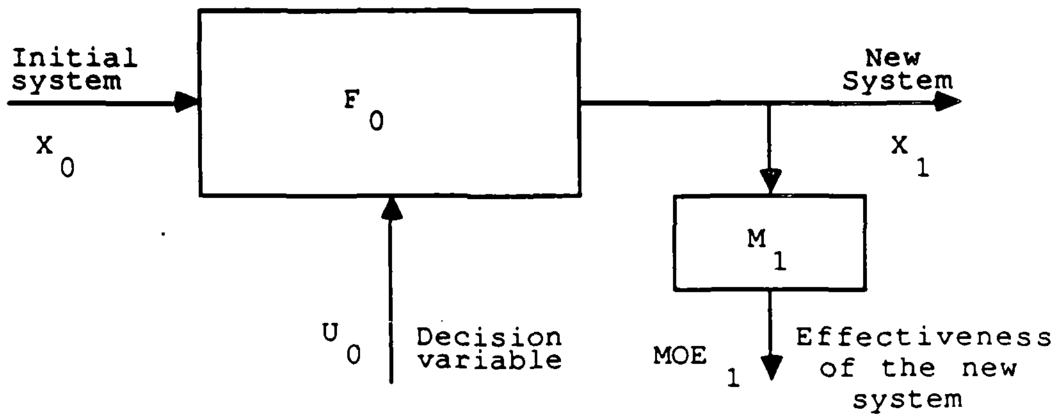


Fig.4.1: Single Step system evolution

To the new system X_1 is associated a measure of effectiveness MOE_1. One can write the following evolution equations:

$$X_1=F_0(X_0,U_0)\text{ , and} \tag{4.1}$$
$$MOE_1=M_1(X_1) \tag{4.2}$$

where F_0 is the function that describes the transformation process, and M_1 the function determining the MOE for X_1. The process sketched out above may be viewed as a single stage evolution.

4.2.2 Multi-stage evolution

One can consider a system that goes through n of the evolution stages described above (Fig.4.2).

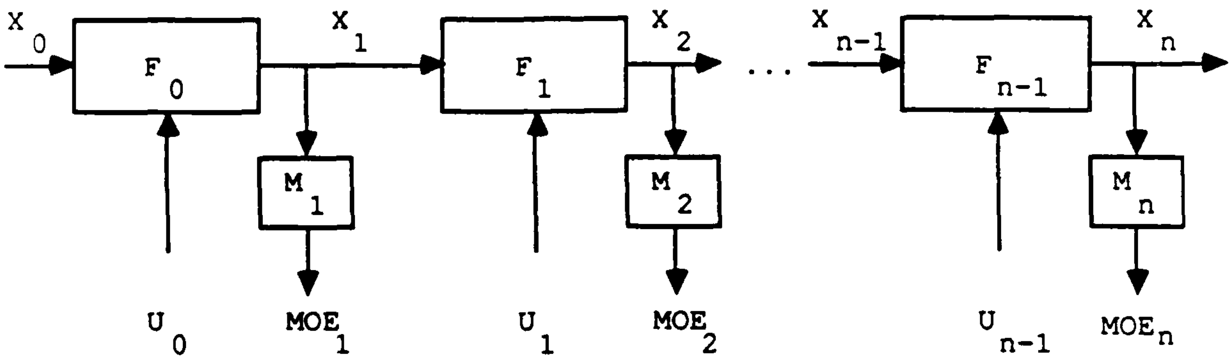


Fig.4.2: Multiple stage evolution

For $t \in [t_i, t_{i+1})$, the system is non-evolving: for each i, t_i is defined as being the time when the system evolves in the sense of chapter 2. For $t \in [t_i, t_{i+1})$, $X_i(t)=X_i$, and the following relations can be deduced from the previous subsection (equations 4.1 and 4.2):

$$X_i=F_{i-1}(X_{i-1},U_{i-1}) \text{, and} \tag{4.3}$$
$$MOE_i=M_i(X_i) \text{ for } i=1...n . \tag{4.4}$$

The functions F_i are fixed because a given initial state and a given decision yield a known outcome. The MOE_i are global measures of effectiveness; They are functions of partial MOE's:

$$MOE_i(X_i)=u_i[E_1(X_i),E_2(X_i),...,E_p(X_i)]=M_i(X_i), \tag{4.5}$$

where $E_1,...,E_p$ are partial MOEs. As the modification process goes on, one may add new partial MOEs or consider a new aggregation of the existing partial MOEs: In general $u_i \neq u_j$, and $M_i \neq M_j$ for $i \neq j$.
The overall goal of the modification process is to maximize

$$V=a_1M_1(X_1)+a_2M_2(X_2)+...+a_nM_n(X_n) \quad (4.6)$$

where $a_1a_2... a_n$ are weighting coefficients, where n is given, and where the decision variables are constrained to be in a specific set. The initial system is given: One must find the sequence of modification that maximizes V.

4.3 DYNAMIC PROGRAMMING

In this section, a dynamic programming algorithm is applied to the multistage decision process presented above.

4.3.1 First step

Suppose that the (n-1)-stage process $(X_1,X_2,...,X_{n-1})$ has been optimized; Then, X_{n-1} is given and the maximization of V is equivalent to the maximization of $a_nM_n(X_n)$. Therefore, the problem is reduced to a single maximization. Given X_{n-1}, find U_{n-1} from the given set of decisions such that:

$$v_{n-1}(X_{n-1})=\text{Max}U_{n-1}\{a_nM_n(X_n)\} = \text{Max}U_{n-1}\{a_nM_n[F_{n-1}(X_{n-1},U_{n-1})]\}. \quad (4.7)$$

To solve this problem, one can apply the methodology developed by Karam [1985] for a single stage process: indeed, Karam considered a single stage process with a decision variable constrained to be in a specific set: for each stage, we have to solve a problem of that kind.

4.3.2 Remaining steps

Suppose that the (n-k)-stage process $(X_1,X_2,...,X_{n-k})$ has been optimized; Then X_{n-k} is fixed and the problem becomes: find U_{n-k} in the given set of decisions such that

$$v_{n-k}(X_{n-k})=\text{Max}U_{n-k}\{B(U_{n-k})+v_{n-k+1}(X_{n-k+1})\} \quad (4.8)$$

$$\text{with } B(U_{n-k})=a_{n-k+1}M_{n-k+1}[F_{n-k}(X_{n-k},U_{n-k})] \quad (4.9)$$

$$\text{and } X_{n-k+1}=F_{n-k}(X_{n-k},U_{n-k}) \quad (4.10)$$

Once again it is a single variable maximization problem, and one can apply the methodology developed by Karam.

4.3.3 Summary

The n variable maximization process is reduced to n single variable problems. By setting i=n-k, one can write:

$$v_i(X_i)=\mathrm{Max}U_i\{B(U_i)+v_{i+1}(X_{i+1})\} \tag{4.11}$$
$$\text{with } B(U_i)=a_{i+1}M_{i+1}[F_i(X_i,U_i)] \tag{4.12}$$
$$\text{and } X_{i+1}=F_i(X_i,U_i) \quad i=0,\dots,n-1 \text{ and } X_0 \text{ fixed.} \tag{4.13}$$

The solution of this problem yields the optimal sequence of decision variables $U_1,U_2,\dots,U_n$, each decision being element of a given set.

4.4 DYNAMIC PROGRAMMING AS PATH DETERMINATION

In this section we present a graphical interpretation of the algorithm developed above; to do so we will apply the procedure to a simple example.

Let us consider a three stages evolution process (n=3) for an initial system X_0; the decision is constrained to be either u_1 or u_2. We assume that the goal is the maximization of $V=MOE_1+MOE_2+MOE_3$.

The dynamic programming algorithm can be represented as a tree (Fig.4.3) where each node corresponds to a given state of the system, and each branch corresponds to a particular decision.

In Fig.4.3, the values of the MOEs for each state of the system are shown on the branches; in this figure, for each node, the upper branch corresponds to decision u_1 and the lower branch corresponds to decision u_2. To find the optimal path on Fig.4.3, we start from X_2 and choose the path leading to a state X_3 with the maximal MOE: we do that for each initial X_2. Then, to determine the best decision to go from X_1 to X_2 we choose the path such that MOE_3+MOE_2 is maximal along that path. Finally, to determine the best decision to go from X_1 to X_2, we choose the path such that $MOE_3+MOE_2+MOE_1$ is maximal along that path. The optimal sequence of decision is, for the example of Fig.4.3: u_1, u_1, u_2; it corresponds to the maximal value for the MOE sum: 1.9

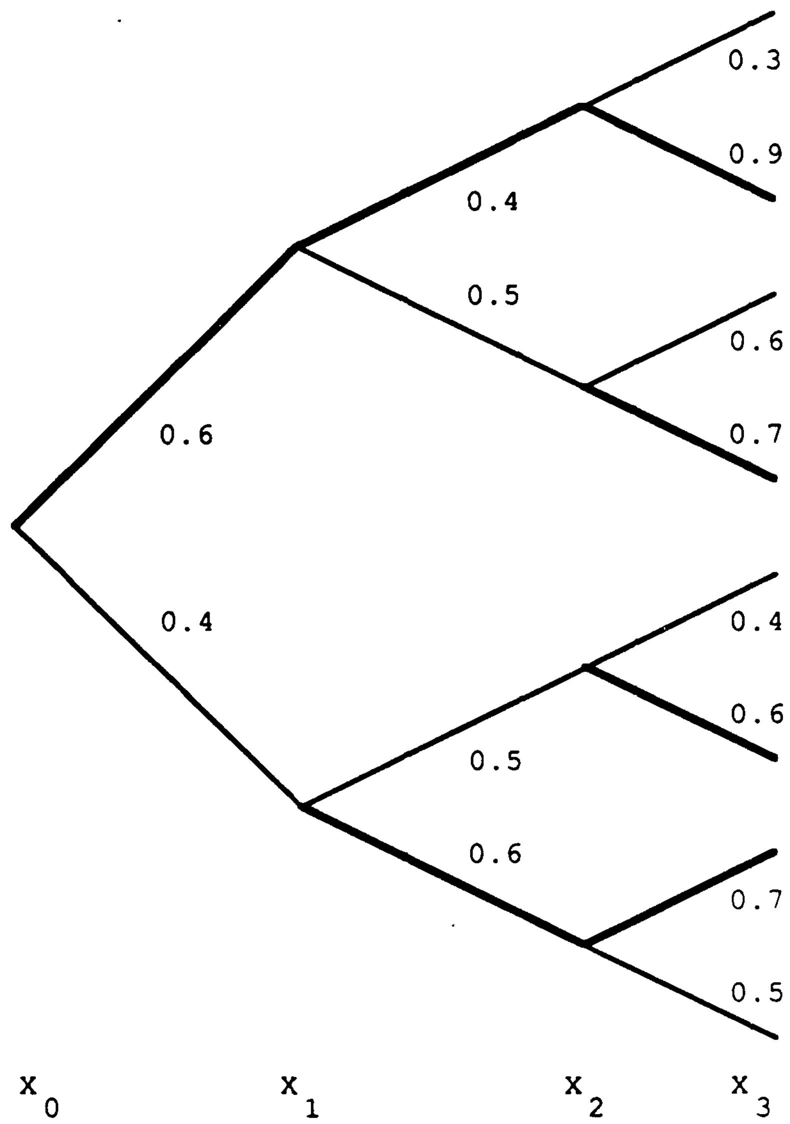


Fig.4.3: A simple example of dynamic programming

In the example considered above, one should note that after choosing decision u_1 to go from X_0 to X_1, it is optimal to choose u_1 and not u_2 as the second decision, although the effectiveness coorresponding to u_1 (0.4) is lower than the one corresponding to u_2 (0.5): the reason for this is that by accepting a temporary degradation of the system, one can reach a high effectiveness state after the third decision; this high effectiveness state (0.9) makes up for the temporary loss at stage 2.

4.5 CONCLUSION

In this chapter a dynamic programming algorithm has been presented: it allows one to determine an optimal sequence of improvement for an evolving system. This algorithm will be applied to a real example in this thesis; the focus of the next chapter is the presentation of this example.

CHAPTER 5
PRESENTATION OF AN AIR DEFENSE SYSTEM

5.1 INTRODUCTION

In this chapter, a large scale system is presented; it will be used later as an example to illustrate the general approach of the last three chapters. This illustration will be provided by a military air defense system known as "Identification Friend Foe Neutral" (IFFN).

5.2 GENERAL BACKGROUND

During the 1970's studies conducted by NATO members and the US Defense Science Board (DSB) diagnosed problems that existed with the aircraft identification (ID) process within NATO. It stimulated research linked to the ID process in real time combat situations; One of the main result was to warn the Air Defense community against relying on a single means for ID, and to emphasize the importance of relying upon many systems to distinguish true friends and neutrals from true foes. Indeed, in modern warfare, because of the extended ranges of missiles, it appears that the reliance on indirect identification is crucial. With this new doctrine, a firing unit may be assigned to a specific target without having ever seen this target. It raises a new problem known as the "Indirect ID" problem: The identification process must be reliable enough to prevent too many friends or neutrals to be engaged and eventually killed.

To assess the effectiveness of the indirect ID process, a test-bed was started in 1979. This project is known as the "Identification Friend Foe or Neutral Joint Test and Evaluation program" (IFFN JT&E) [IFFN Test Plan 1985]. The purpose of the IFFN JT&E is "to assess baseline capabilities within the air defense C^2 structure to perform the IFFN function, to identify deficiencies in the performance of that function, and to define potential near-term procedural and equipment modifications for further testing".

Therefore, the test-bed is a complete surrogate of the actual system which, of course, can hardly be tested. By conducting experiments on the IFFN JT&E facilities it is

expected that the effectiveness of the actual system can be assessed. Then, the test-bed should also provide a means to improve the real system by adding or changing components, structures, or doctrines and to evaluate the resulting effectiveness. In what follows no distinction will be made between the test-bed and the real system: indeed, it is assumed that the test facility is an excellent model of the actual system.

As one should note, the approach followed in the development of this large scale system fits exactly in the framework of chapter 3 where a methodology to assess the effectiveness of real systems has been presented. In the remaining of this chapter, the actual IFFN system and a simplified mathenatical model of this system are presented.

In the next section the actual system is presented; then, a simplified mathematical model will be introduced.

5.3 DESCRIPTION OF THE ACTUAL SYSTEM

The actual system we will focus on is an air-defense system operating in the Central region of Europe. The overall mission of the system is to defend a specified airspace from an air attack carried out by enemy aircraft and missiles.

This system is shown in Fig.5.1; it is composed of C^2 nodes that coordinate the action of weapons; these nodes are represented in Fig.5.1, and can be classified into three major groups: Fire Directing Centers (FDC), Control Reporting Centers (CRC), and a third group composed of more specific nodes such as databases and higher level nodes. Their role is to coordinate the weapons shown in Fig.5.1; these weapons can be divided into two categories: Surface to Air Missiles (SAM) such as Hawks and Patriots units, and fighter such as F-15 and F-16 aircraft.

The mission of the system is to engage and destroy hostile airborne targets or otherwise deny the enemy access to the defended airspace: in particular, the enemy aircraft must be stopped before they can fire missiles at friendly assets. The system must be selective enough to minimize killing friends (F-15 and F-16) or neutrals such as commercial aircraft that are assumed to be flying in the Central Region at the time of the battle. To accomplish this mission, the air-defense system must perform a number of sequential functions which define the air defense process; this process is characterized by six primary phases: detection, tracking, identification (ID), allocation, engagement, and target kill.

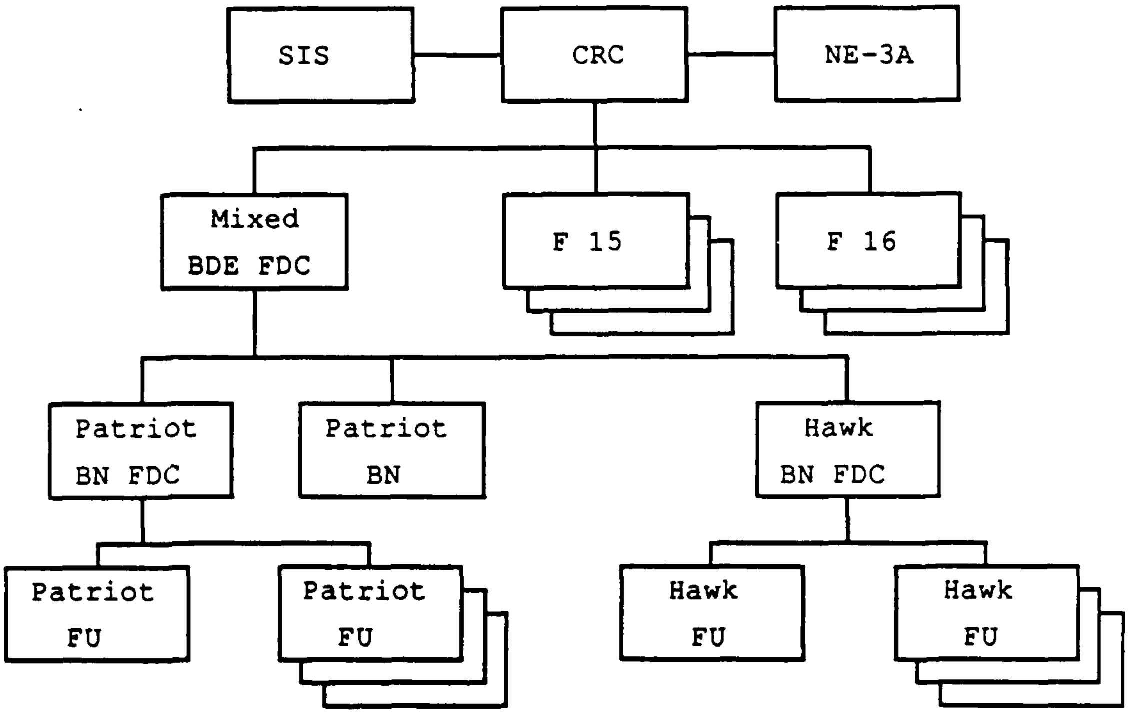


Fig.5.1: Structure of the actual IFFN system

Abreviations:

NE-3A: NATO airborne early warning sytem (Boeing 707); this is a high altitude detection aircraft.

SIS: Special Information System; this is a source of intelligence information available to basic nodes of the system (CRC or Control and Reporting Center nodes).

CRC: Control and Reporting Center; this C^2 node is responsible for the overall coordination of the system.

FDC: Fire Directing Center; these C^2 nodes are responsible for the coordination of a battalion.

FU: Firing Unit

BN: Battalion

BDE: Brigade

In this complex system, the missiles fired either by a fighter or by a SAM unit are Beyond Visual Range (BVR) weapons: a firing unit does not see the targets it is shooting at; the fire parameters are given to this firing unit by the C^2 system on the basis of identification performed by other units called "detecting units". This indirect identification

process justifies the C^2 structure that lies above the weapons in Fig.5.1: it is the responsibility of this structure to pass correct and accurate parameters to firing units.

In the next section, a simplified model of this complex system is introduced; the remaining of the chapter will develop this simplified model.

5.4 SIMPLIFIED MODEL: AN OVERVIEW

In the simplified model, we are going to focus on, the enemy forces are assumed to consist of aircraft only; these enemy aircraft seek to enter the friend's territory; they can fire Air to Surface Missiles (ASM) and Air to Air Missiles (AAM) in order to destroy both ground units and airborne units. For their defense, the friends have aircraft that can fire AAM, and ground units that can fire Surface to Air Missiles (SAM) in order to kill the invading aircraft. An enemy unit will refer to aircraft; a friendly asset will refer to both aircraft and ground firing units; for the neutrals, a unit will refer to an aircraft (commercial aircraft).

In this model, the geometry has been simplified: it consists of a straight line (FSCL or Fire Support Coordination Line) separating the friendly forces from the enemy: the system under consideration lies behind the FSCL, and hostile aircraft are heading towards this line at speed V. This model is represented in Fig.5.2:

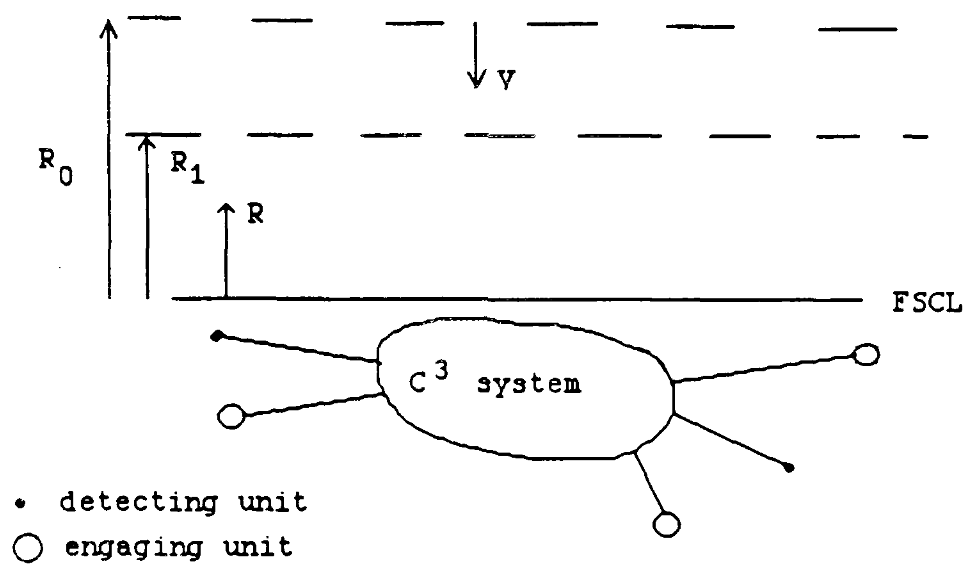


Fig.5.2: Simplified IFFN model

R_0 is the measurement volume of the system
R_1 is the range of the enemy's air surface missiles
R is the distance separating the enemy's aircraft from the FSCL
V is the speed of enemy's aircraft
FSCL is the line between friends and enemies

The distance from these aircraft to the FSCL is R. Detecting units are in charge of the detection of aircraft, while engaging units are responsible for engaging them: an aircraft will be detected by a given detecting unit and engaged by another unit (engaging unit): the task of the C^3 system is to identify correctly an aircraft and to allocate it to a given engaging unit.

In order to protect friendly assets, the enemy's aircraft must be stopped before they reach R_1, and can fire missiles.

We assume that the friendly aircraft as well as the neutral aircraft are flying at any speed and in any direction in the diagram sketched above.

In what follows, this simplified model is developed, based on the work carried out by Logicon [Logicon 1986]; throughout the chapter, the notation defined in the IFFN documents [IFFN Test Plan 1985, Logicon 1986] will be used.

5.5 PARAMETER DEFINITION

To carry out the effectiveness analysis of this system, one must first identify the parameters (the independent variables). The parameters we are going to consider are those defined in the IFFN documentation [IFFN Test Plan 1985, Logicon 1986]; in order to design tests to run on the system, a test-matrix has been defined; this test matrix consists of two sub-matrices: a "basic matrix" (Fig.5.3) and a "test excursion matrix" (Fig.5.4): in what follows, only one activity level will be considered. The parameters are the quantities that are varied to go from one element of the test excursion matrix to an other. Each element of the basic matrix is called a basic operating point; each basic operating point associated with all the possible test excursions is called a test-cell: therefore, to each basic operating point is associated a test cell. In the following analysis, we are going to consider one test cell at a time, or one element of the basic matrix at a time: thus, only the test excursion matrix is going to be considered.

Variables \ Scenario	Activity level 1	Activity level 2
Nominal performance		
Maximum system ID performance		
Remove ACP		
Remove electronic ID		
Remove Q&A IFF nodes		
Remove JTIDS		
Remove direct positive hostile		
Centralized mode of control		
Remove NE-3A		
Remove SIS		

Fig.5.3: Basic matrix

Abreviations:

JTIDS: Joint Tactical Information Distribution System; this is a high capacity information distribution system providing secure, flexible and jam resistant information transfer among the nodes of the C^3 system.

NE-3A: NATO airborne early warning sytem (Boeing 707); this is a high altitude detection aircraft.

SIS: Special Information System; this is a source of intelligence information available to Control and Reporting Center (CRC) nodes.

ACP: Air Control Procedure: this is the set of rules and protocols that enable the system to monitor efficiently the airspace.

Q&A IFF nodes: Question and Answer devices for Identification Friend Foe; these are devices located at the weapon level that provide local ID information for a given weapon.

Variables \ Scenario	Activity level 1	Activity level 2
SIS into CRC		
Variant ACP		
Perfect ID		
Decentralized mode of control		
Improved Q&A IFF devices		

Fig.5.4: Test Excursion matrix

For the sake of the analysis to come, the parameters will be:

- time needed to pass information between two nodes (P_1)
- range from aircraft to FSCL at time of detection (P_2)
- quality of identification (P_3)
- level of centralization of control (P_4)
- quality of target allocation and engagement (P_5)

These parameters correspond to the test excursion matrix; they are independent variables included in [0,1]:

P_1 corresponds to the SIS (Special Information System) included or not into the CRC (Control and Reporting Center); this fact is modeled by a varying the time delay required to pass information between two nodes.

P_2 corresponds to the variable ACP (Air Control Procedure): the ACP is the set of rules that enables the system to monitor the air space; the effect of varying the ACP is assumed to be the variation of the range from aircraft to FSCL at time of detection: the better the ACP, the larger this range.

P_3 corresponds to the variation in the quality of identification (ID).

P_4 corresponds to the level of centralization.

P_5 corresponds to the quality of the Q&A IFF devices (Question and Answer devices for Identification Friend Foe): since Q&A IFF devices provide local ID information at the weapon level, the quality of these components has a direct effect on allocation and engagement performances.

These parameters reflect the experiments that will be conducted on the IFFN

testbed: basic operating points are represented by the basic matrix; from these basic operating points, excursions will be considered. These excursion will be conducted by varying the parameters defined above. To each basic operating point corresponds a "test cell". Only the five parameters defined above will be varied to apply the SEA methodology: other parameters describing either the system or the context will be fixed in the SEA analysis.

5.6 MEASURES OF PERFORMANCE

5.6.1 Introduction

After having defined the parameters, one must specify the MOPs of interest for the system at hand. These MOPs must allow one to make a decision concerning the system: they must have a clear physical interpretation.

5.6.2 Notation

Let us denote:

$x(t)$: number of friends at time t

x_0: initial number of friends

$y(t)$: number of enemies at time t

y_0: initial number of enemies

$z(t)$: number of neutrals at time t

z_0: initial number of neutrals

$n(t)$: fraction of friendly forces lost at time t

$m(t)$: fraction of enemy forces at time t

$$n(t)= (x_0- x(t)) /x_0 \ , \quad m(t)= (y_0- y(t))/y_0 \qquad (5.1)$$

Quantities x_0,y_0,z_0 are measured at the initial time: that is the time when hostile aircraft enter the detection volume (R_0) of the system.

Quantities $x(T_f),y(T_f),z(T_f)$ are measured at the final time T_f. The battle stops when either the friendly forces or the enemy's have lost a given fraction of their assets. The final time T_f is defined by

$$(n(t) < n_f \text{ and } m(t) < m_f \text{ for all } 0 \leq t \leq T_f),$$

and $$(n(T_f) = n_f \text{ or } m(T_f) = m_f) \tag{5.2}$$

where n_f and m_f represent the strategy of each side: since the friendly forces are defending their own territory. they are probably willing to loose a greater fraction of their forces than the enemy: $n_f > m_f$ is very likely to be true.

5.6.3 MOP Definition

In order to enable one to evaluate the system, the MOPs we will consider must have a clear physical meaning: as pointed out in section 5.3, we want the system to perform a threefold task: deter enemy from entering the friend's territory, stop the enemy as far as possible from this territory and before it can fire missiles aimed at friendly assets, and kill as few neutral as possible.

To evaluate the first task, we need a quantity indicating whether the friends win the battle or not; an indicator of the willingness of the friends to keep on fighting is the ratio

$$x(T_f)/(x_0*(1-n_f)) = (1-n)/(1-n_f). \tag{5.3}$$

This ratio measures how far the remaining forces of the friends are from their lowest acceptale level as given by $1-n_f$. If the friends win the battle, this ratio will be greater than one; if they loose, it will be equal to one since they are giving up when their level of losses reaches n_f. A similar ratio can be defined for the enemy. Then, we will consider as the first MOP of our problem, the ratio of these two ratios: this ratio of ratios will compare the willingness of the two opponents to keep on fighting. Thus we define

$$\text{MOP1} = [x(T_f)/(x_0*(1-n_f))]/[y(T_f)/(y_0*(1-m_f))] \tag{5.4}$$

The second quantity we want to evaluate is the number of neutral aircraft killed by the friends; indeed, since we are interested in evaluating the friend's system as opposed to he enemy's one, we consider only the neutrals shot down by the friend's air-defense. For simplicity, we assume that the enemy is not killing any neutral, and that, if a neutral is killed, it is by a friend. Then, we are interested in the number of neutrals remaining at the end of the battle; the MOP that will measure this performance of the system is:

$$MOP2 = z(T_f)/z_0 \tag{5.5}$$

The last quantity of interest is the distance of the enemy from the FSCL when the battle ends. This is measured by the following ratio:

MOP3 = (Distance of the remaining enemy's aircraft at time T_f)/R_1

5.6.4 Interpretation of the MOPs:

If MOP1>1, the hostile aircraft fly back because their level of losses has reached m_f while $n(T_f)<n_f$; indeed, if MOP1>1, then

$$[x(T_f)/(x_0*(1-n_f))]>[y(T_f)/(y_0*(1-m_f))] \tag{5.6}$$

and one of the two terminating condition is true

$$1-y(T_f)/Y_0 = m_f \quad \text{or} \quad 1-x(T_f)/x_0 = n_f \; . \tag{5.7}$$

Let us suppose that the latter of these two equalities is true. Then

$$y(T_f)/(y_0*(1-m_f)) < 1 \tag{5.8}$$

which is equivalent to

$$(y_0-y(T_f))/y_0 > m_f \; . \tag{5.9}$$

Therefore, if

$$MOP1 > 1 \quad \text{and} \quad 1-n_f = x(T_f)/x_0 \; ,$$

then the level of loss for the enemy at T_f is greater that its maximum: it is impossible. The conclusion is that, if

$$MOP1 > 1 \; , \quad \text{then} \quad 1-m_f = y(T_f)/Y_0 \tag{5.10}$$

and the attack has been repulsed.

If MOP1<1, the friendly forces give up and loose the battle because their level of losses has reached n_f while $m(T_f)<m_f$: the proof is the same as the one above.

MOP2 has a straightforward interpretation: since we assume that only the friends are killing neutral assets, MOP2 is the fraction of neutral aircraft remaining at the end of the battle.

MOP3 represents a ratio of distance to the FSCL

If MOP3>1, the enemy aircraft are stopped before they can fire missiles aimed at friendly assets.

If MOP3<1, the enemy aircraft can fire missiles before being stopped. The greater this ratio is, the better it is for the friendly forces.

5.7 MAPPING FROM THE PARAMETER SPACE INTO THE MOP SPACE

5.7.1 Introduction

In this part we are going to map the parameter space into the MOP space; the basis of this section is the work completed by Logicon [Logicon 1986], where about thirty quantities representing the way the IFFN system performs are defined. Most of these quantities are conditional probabilities that describe the different stage of the air-defense process. These quantities are called MOPs and MOEs in the Logicon documentation; since they are different from the MOPs and MOEs we consider for the SEA methodology, and since we want to keep the notation defined by Logicon, these quantities will be denoted "Mop" and "Moe" as opposed to "MOP" and "MOE" in the SEA methodology.

First of all, we are going to aggregate the conditional probabilities (Mops and Moes) defined in the IFFN documents [Logicon 1986] in a smaller number of quantities; then, we will determine the values of these aggregate quantities in terms of the independent variables of the problem (parameters); finally, we will use a Lanchester model [Echkian 1982, Taylor 1974, Moose and Wozencraft 1983] to derive the MOPs (for our problem) in terms of the parameters.

5.7.2 Aggregation of the IFFN Mops and Moes

The IFFN documents define conditional probabilities (Mops and Moes) that are useful for setting up a mathematical model of the system. These quantities correspond to stages in the ID process; these stages have been described in the IFFN documentation [Logicon, 1986], and in Gandee (1986). The purpose of this section is to aggregate these stages in order to determine four basic quantities: probability of engaging, a friend, a neutral, or an hostile, and the time elapsed between detection and engagement.

The basic stages of the model are: detection, identification (ID), comparison between different IDs coming from different detecting units, conflict resolution, allocation, and engagement. A conditional probability is associated to each of these stages. To these six stages corresponding to physical processes, one must add a fictitious step that describes the probability of true identification. With a decision tree [White 1985], it is then possible to determine three basic quantities: probability to engage a friend, a neutral, or an enemy.

The decision trees are presented in figures 5.5, 5.6, and 5.7; each of these trees is conditioned on the true ID of an aircraft: for example, on the first tree (Fig.5.5), all the probabilities assume that the true ID of the aircraft under consideration is "friend". Let us now review the stages of these trees with the notation defined in the IFFN documents [IFFN TestPlan 1985, Logicon 1986]:

-"Detection": probability of detecting an aircraft (Moe1)

-"ID": probability of identifying (ID) an aircraft (Moe3)

-"ID = True": this is a fictitious physical process that allows one to take into account the possibility of wrong ID in the decision trees. Because of the three different types of aircraft, there are three possible outcomes for this stage: ID as a friend, a neutral or a hostile. In the decision trees appear the quantities P(ix/yd): probablity of identifying an aircraft as x, given its true identity is y and it has been detected, where x and y can be "friend" (f), "neutral" (n), or "hostile" (h)

-"ID passed = ID recorded": represents the comparison between IDs coming from different detecting units: Mop3.5 is the probability that there is no conflict.

-"Conflict resolution": in case of conflict between different IDs, this stage represents the probability of resolving this conflict (Mop3.6).

-"Allocate": represents the allocation process. Mop4.1 has to be broken up into three different probabilities for the three trees considered:

P(a/xi): probability of allocating an aircraft, given its true ID is x and it has been identified, where x can be "friend" (f), "neutral" (n), or "hostile (h).

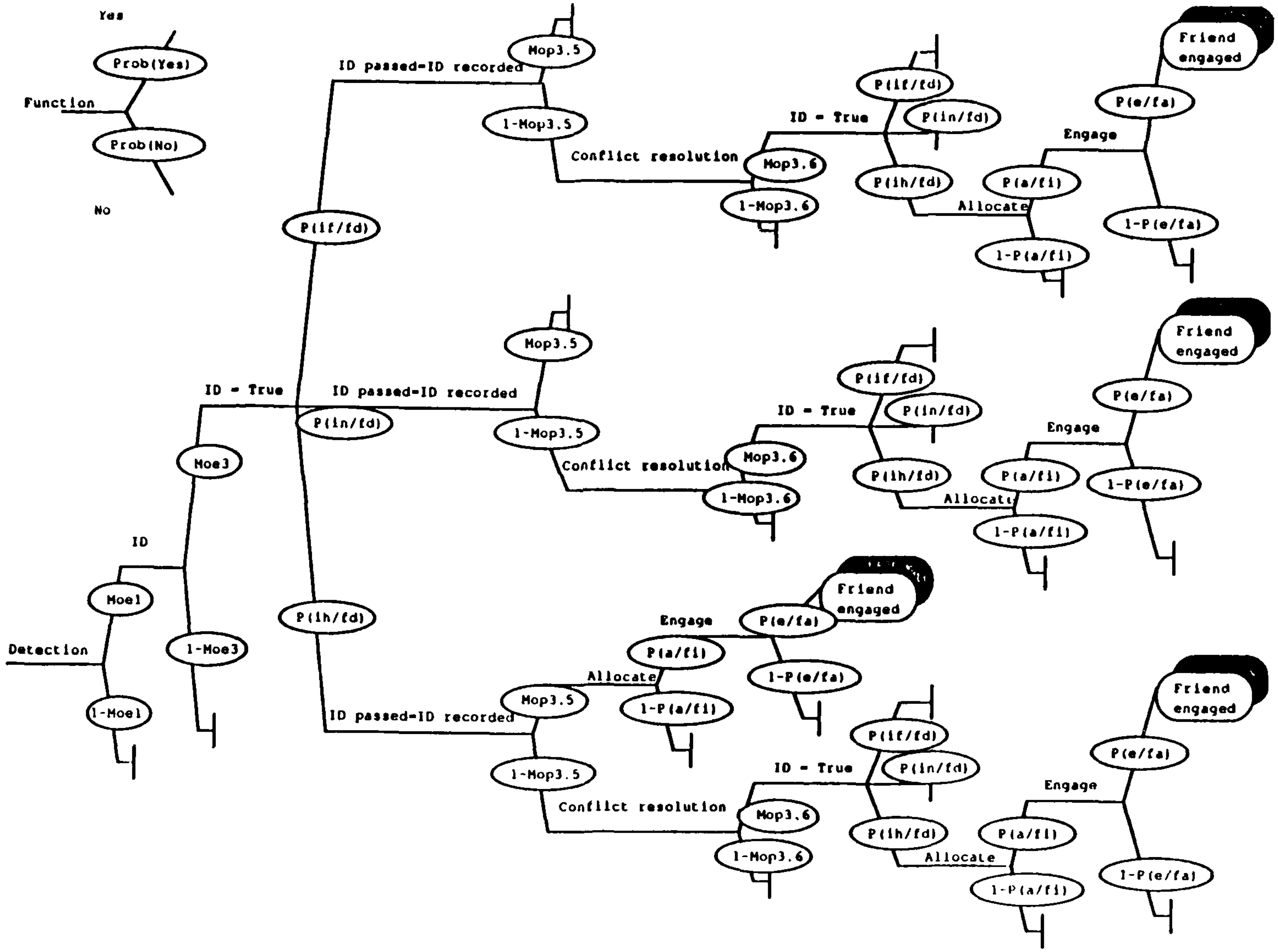


Fig.5.5: Decision Tree for Friends

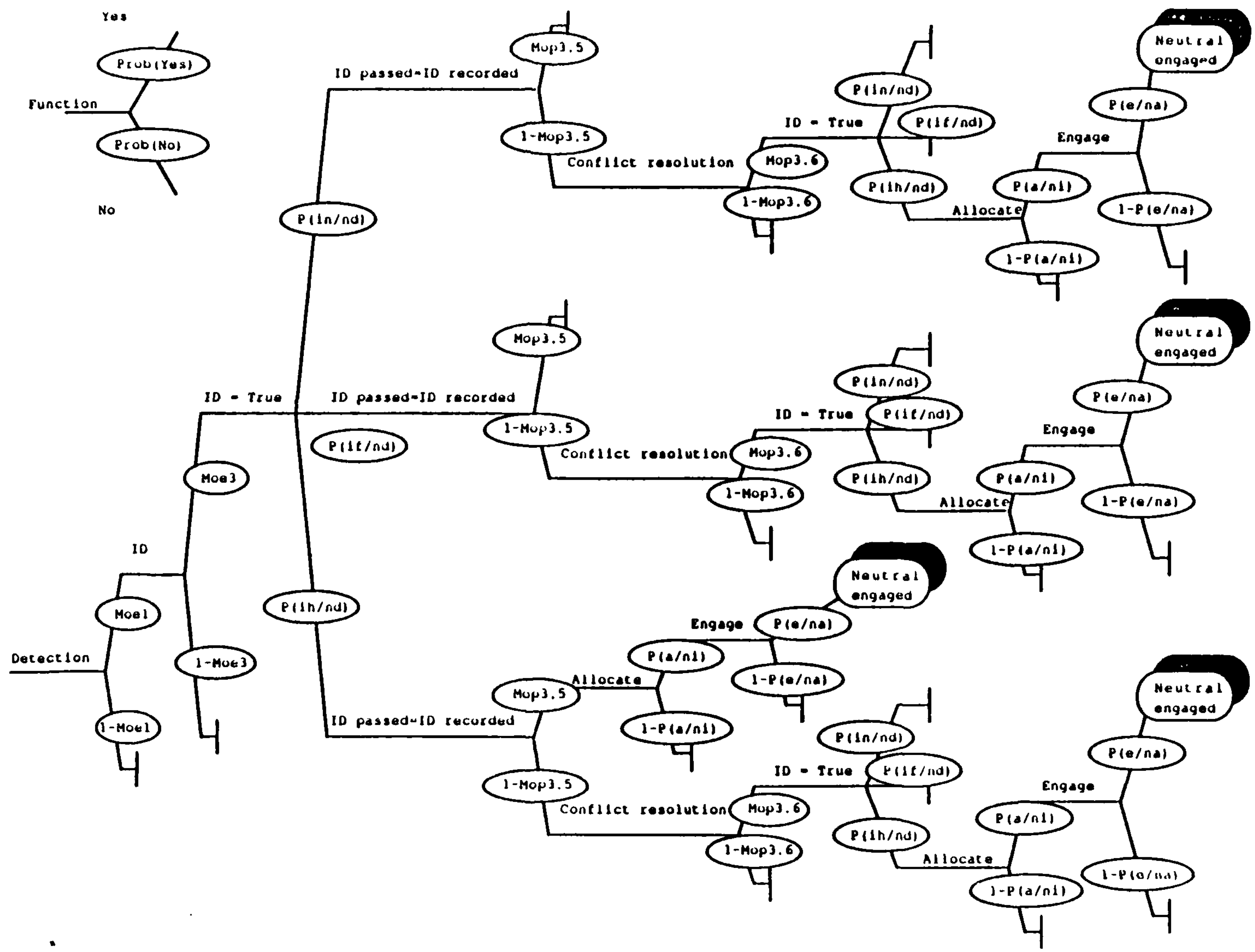


Fig.5.6: Decision Tree for Neutrals

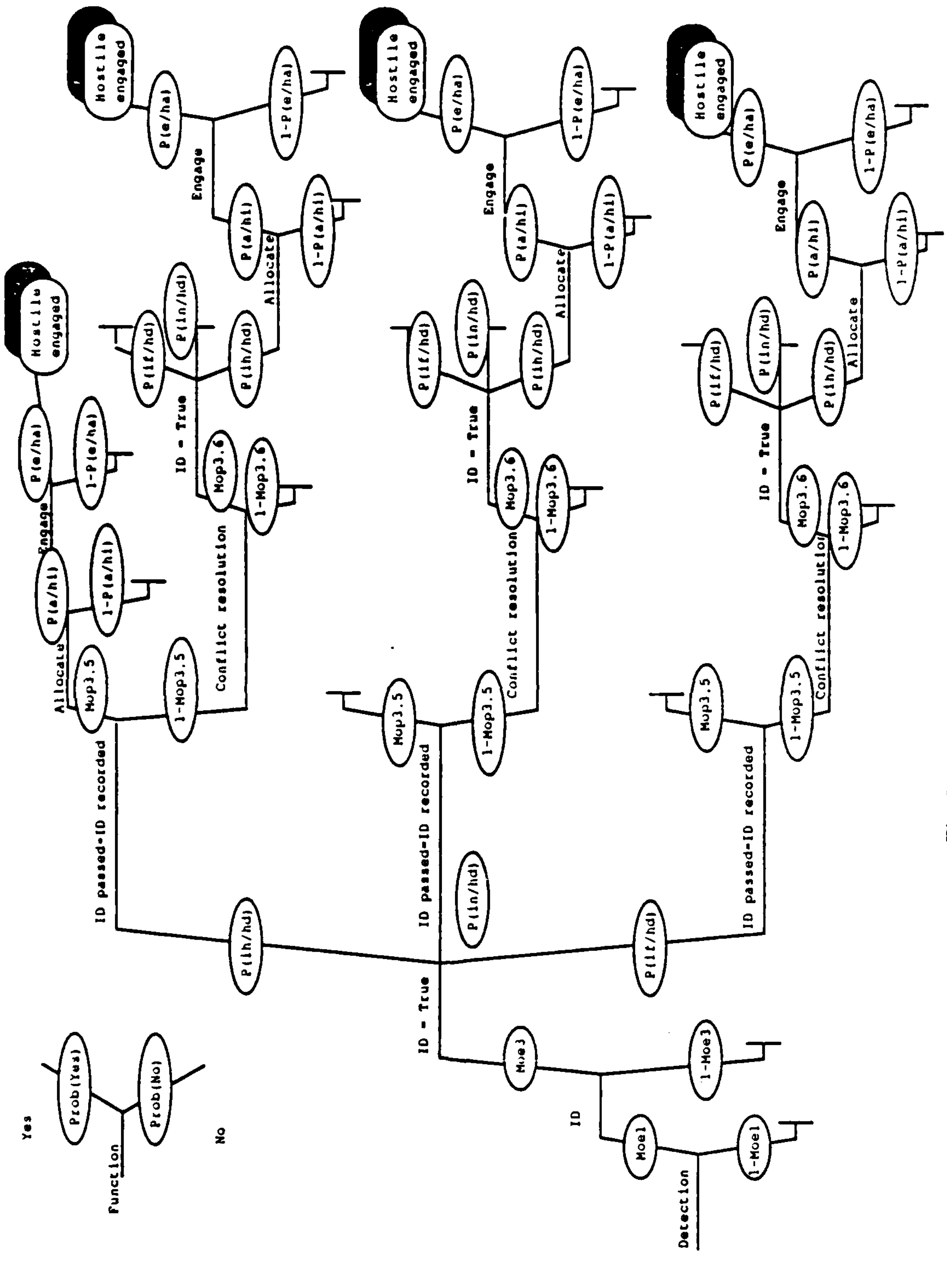


Fig.5.7: Decision Tree for Hostiles

-"Engage": represents the engagement process. Mop5.1 has to be broken up into three different probabilities for the three trees considered:

P(e/xi): probability of engaging an aircraft, given its true ID is x and it has been identified, where x can be "friend" (f), "neutral" (n), or "hostile (h).

Some simplifications have been made in the work completed by Logicon [Logicon 1986]:

-since we are interested only in the indirect ID problem, we do not take into account the "local threat evaluation" stage defined in the IFFN documentation.

-since the tracking stage defined in Logicon [1986] doesn't affect the probabilities considered in our model, this stage has been removed from the analysis.

-only a two stages verification process is considered, instead of a multi-stage one.

-the stages "ID passed" and "ID passed = ID recorded" defined in Logicon [1986] have been aggregated in a single stage ("ID passed = ID recorded").

These decision trees enable one to determine three basic quantities: probability to engage a friend (Moe7), a neutral (Moe8), or a hostile (Moe9). Since

$$P(ih/fd) + P(in/fd) + P(if/fd) = 1\ , \tag{5.11}$$

$$P(ih/nd) + P(in/nd) + P(if/nd) = 1\ , \tag{5.12}$$

$$P(ih/hd) + P(in/hd) + P(ifhfd) = 1\ , \tag{5.13}$$

one can deduce:

$$Moe7 = Moe1 * Moe3 * P(ih/fd) * P(a/fi) * P(e/fa) * A \tag{5.14}$$

with $A = (1-P(ih/fd)) * (1-Mop3.5) * Mop3.6 + Mop3.5 + P(ih/fd) * (1-MOP3.5) * Mop3.6 \quad ;$

$$Moe8 = Moe1 * Moe3 * P(ih/nd) * P(a/ni) * P(e/na) * B \tag{5.15}$$

with $B = (1-P(ih/nd)) * (1-Mop3.5) * Mop3.6 + Mop3.5 + P(ih/nd) * (1-MOP3.5) * Mop3.6 \quad ;$

$$Moe9 = Moe1 * Moe3 * P(ih/hd) * P(a/hi) * P(e/ha) * C \tag{5.16}$$

with $C = (1-P(ih/hd))*(1-Mop3.5)*Mop3.6 + Mop3.5$
$+ P(ih/hd)*(1-Mop3.5)*Mop3.6$;

It is also possible with these decision trees to determine the "mean time elapsed between detection and engagement" (Moe10).

$$Moe10 = Mop2.3 + Mop3.4 + Mop4.4 + Mop5.3$$

where (Mop2.3+Mop3.4) is the time between detection and ID, Mop5.3 is the time elapsed between allocation and engagement, and Mop4.4 is the time elapsed between ID and allocation [Logicon 1986]: only this latter time is assumed to vary, depending on whether there is a conflict during the ID process or not. Therefore, with these assumptions,

$$Mop2.3 + Mop3.4 + Mop5.3 = \text{constant}, \quad (5.17)$$

$$Mop4.4 = Mop3.7+(1-Mop3.5)*Mop3.7 \quad (5.18)$$

and

$$Moe10 = Mop2.3 + Mop3.4 + Mop4.4 + Mop5.3 \ . \quad (5.19)$$

5.7.3 From the parameters to the aggregate quantities

In this section we are going to express, the Mops and Moes considered in the previous section in terms of the five parameters.

Mop1.2 (distance from aircraft to FSCL at time of detection) is assumed to be affected only by the variation of the Air Control Procedure (ACP): the ACP is the set of rules that enable the system to monitor the air-space. The better the ACP, the greater Mop1.2 is: indeed, with a good ACP, one is able to detect an enemy within a very short period of time; it yields a large Mop1.2 because within this time period, the aircraft cannot go very far inside the detection volune. We assume:

$$Mop1.2 = R_0 * P_2 \qquad 0 \leq P_2 \leq 1 \quad (5.20)$$

The probabilities corresponding to MOP3.1 are assumed to be only functions of P_3. If P_3 equal to 1 corresponds to perfect ID capabilities, we have the following relations:

$$P(ih/hd) = P_3 \ , \quad (5.21)$$

$$P(ih/nd) = P(ih/fd) = 1-P_3 \ , \ 0 \leq P_3 \leq 1 \quad (5.22)$$

The probabilities Mop3.5 (ID passed = ID recorded) and Mop3.6 (conflict resolution) are assumed to be only function of the level of centralization. The probability of conflict (Mop3.5) must decrease, and the probability of conflict resolution (Mop3.6) must increase when the process becomes more centralized ($P_4=1$ when centralization is total).

$$\text{Mop3.5} = 0.75 - 0.5 * P_4 \tag{5.23}$$
$$\text{Mop3.6} = 0.25 + 0.5 * P_4 \qquad 0 \leq P_4 \leq 1 \ . \tag{5.24}$$

The time needed to exchange ID information between a unit that produces this information and another that uses it (Mop3.7) is assumed to be a function of the level of centralization (P_4) and of the integration of the Special Information System (SIS) into the Control and Reporting Center (CRC) (P_1). If the SIS is included into the CRC ($P_1=0$), Mop3.7 decreases; if the process is centralized, Mop3.7 increases: these assumptions are justified by the two following remarks: to exchange ID information between two given components, the greater the level of centralization (the greater P_4), the greater the number of information exchanges between nodes is, and the greater the time is; if the SIS is included into the CRC (P_1 decreases), an important database is included into a node of the system, thus reducing the overall time required for exchanging ID information. If we assume that P_1 and P_4 are weighted evenly, and that the time to exchange information cannot be smaller than $(\text{Mop3.7})_{max}/2$, we have the following relation:

$$\text{Mop3.7} = (\text{Mop3.7})_{max} * (0.5 + 0.25 * P_1 + 0.25 * P_4) \ , \tag{5.25}$$

where $(\text{Mop3.7})_{max}$ is a constant and $0 \leq P1 \leq 1$.

The probabilities of allocation and engagement are assumed to be only functions of P_5. With improved Question and Answer devices for Identidication Friend Foe (Q&A IFF devices), that is if P_5 increases, P(a/hi) and P(e/ha) are going to increase, while P(a/fi), P(e/fa), P(a/ni), and P(e/na) are going to decrease. This is justified by the fact that Q&A IFF devices are components that are at the weapon level: if the true ID of an aircraft is friend or neutral, the better the quality of these devices, the lower the probability this aircraft will be allocated and engaged is; if the true ID of the aircraft is hostile, the greater the probability of allocating it and engaging it is. We assume the following relations:

$$P(a/hi) = P(e/ha) = (0.875 + 0.125 * P_5)^{1/2} \tag{5.26}$$

$$P(a/fi) = P(e/fa) = P(a/ni) = P(e/na)$$
$$= (0.125 - 0.125 * P_5)^{1/2}, \quad (5.27)$$
with $0 \leq P_5 \leq 1$.

In the equations of the model only the probabilities of "allocating and engaging" an aircraft are considered: only the products $P(a/xi) * P(e/xa)$, where x is the true ID of the aircraft, are considered; in order to keep these products as linear functions of P_5, an exponent 1/2 has been introduced in expressions 5.26 and 5.27.

We can now express the four aggregate quantities we are interested in (Moe7, Moe8, Moe9, Moe10) in terms of the parameters of the problem. Some constants remain to be fixed (Moe1, Moe3, Mop3.7_{max}, R_0 ...); since these constants are not going to be varied the System Effectiveness Analysis, they are assumed to be given quantities for each basic operating point to which the methodology is applied.

In the next section, we will determine the MOPs of our problem in term of these four quantities.

5.7.4 Lanchester model

The final step to determine the three MOPs we are going to consider is the use of the Lanchester equations [Ekchian 1982, Taylor 1974, Moose and Wozencraft 1983]: from an initial number of aircraft of each type, we determine the final number in each category ($x(T_f)$,$y(T_f)$,$z(T_f)$) on the basis of the probabilities computed with the decision trees; the final time is a crucial issue because the Lanchester models are valid only for a great number of entities: they are aggregate models. The Lanchester equations provide a means to compute the number of friends, foes, and neutrals at each time of the battle; with the variation of the number of friends and hostiles during the battle, one can determine the terminating time T_f using the criteria of equation 5.2; this terminating time enables one to obtain the number of friends ($x(T_f)$), foes ($y(T_f)$), and neutrals ($z(T_f)$) at the end of the battle. These are the three quantities that one needs to evaluate MOP1 and MOP2 (see equations 5.4 and 5.5). The third MOP of our model is then equal to

$$MOP3 = (Mop1.2 - V * T_f)/R_1 \quad (5.28)$$

(Mop1.2 is the range from aircraft to FSCL at time of detection).

In the Lanchester model we are going to consider, we assume that all the friends are within the weapon range of the enemy, and that all the neutrals and all the hostiles are within the weapon range of the friend's units. Since we are interested in the performance of the indirect ID process, we consider losses in friendly forces to be due to the enemy action, and to errors within the friend ID process; we consider losses in the enemy forces, and neutral losses to be due to the friend's fire only: indeed, we are interested in the performance of the friend's air defense system only.

For the firing relation between the friends and the enemy, we assume that each unit is aware of the exact location of the remaining enemy units, so that when a target is destroyed, fire may be immediately shifted to a new target: it means that, if we consider only x (Friend) and y (Enemy) firing at each other without any error, we have:

$$[dx/dt]/y \quad \text{and} \quad [dy/dt]/x$$

be constant over time (x losses are only dependent on the level of forces y, because each y unit knows exactly where to fire at an x unit). Thus, for the interaction between x and y, we have:

$$dx/dt = -b*y \quad ,$$

and $$dy/dt = -c*x \quad ,$$

where b and c are positive constants (Lanchester square law). Since we want to consider errors in the ID process for the friends, we must add a new term in the friend losses: this term is proportional to x and the evolution of x is then described by:

$$dx/dt = -a*x - b*y \; .$$

We assume that the neutral losses are only due to the friend's errors, and that the friend's fire on the neutrals is uniformly distributed over a constant target area; therefore, we are going to use a linear attrition law:

$$dz/dt = -d*x*z \; ,$$

where d is a positive constant. This law means that the rate of destruction of neutral aircraft produced by each unit x depends linearly on z: it is consistent with the expectation that, as the number of neutral aircraft decreases, their rate of destruction decreases also (there will be less noise in the ID process and therefore less error).

Therefore, we have the following three equations:

$$dx/dt = -a*x - b*y \qquad x=x_0 \text{ at } t=0 \tag{5.29}$$
$$dy/dt = -c*x \qquad y=y_0 \text{ at } t=0 \tag{5.30}$$
$$dz/dt = -d*x*z \qquad z=z_0 \text{ at } t=0 \tag{5.31}$$

for $0 \leq t \leq T_f$, T_f determined as shown above (equation 5.2).

"a" is the probability of engaging a friend per unit of time: thus,

$$a = Moe7/Moe10 \tag{5.32}$$

Similarly,

$$c = Moe9/Moe10 \tag{5.33}$$

and, $$d = Moe8/Moe10 \tag{5.34}$$

b is assumed to be exogenous and fixed independently of the parameters: indeed, b is the probability for an enemy to kill a friend; it is an exogenous quantity.

The integration of equations 5.29, 5.30, 5.31 yields:

$$x = ß_1*\exp[-(a+s)*t/2] + ß_2*\exp[-(a-s)*t/2] \tag{5.34}$$
$$y = ß_3*\exp[-(a+s)*t/2] + ß_4*\exp[-(a-s)*t/2] \tag{5.35}$$
$$z = z_0*\exp[d*(y-y_0)/c] \tag{5.36}$$

with, $$s = [a*a + 4*b*c]^{1/2}, \tag{5.37}$$
$$ß_1 = (a+s)*[x_0 - (a-s)*y_0/(2*c)]/(2*s) \tag{5.38}$$
$$ß_2 = (a-s)*[(a+s)*y_0/(2*c) - x_0]/(2*s) \tag{5.39}$$
$$ß_3 = c*[x_0 - (a-s)*y_0/(2*c)]/s \tag{5.40}$$
$$ß_4 = c*[(a+s)*y_0/(2*c) - x_0]/s \tag{5.41}$$

The final time T_f is given by

$$n(T_f) = n_f \tag{5.42}$$

or, $$m(T_f) = m_f \tag{5.43}$$

Since one cannot find a closed expression for T_f, the final time must be computed

numerically.

We can now compute the three MOPs (MOP1, MOP2, MOP3) of the system for each point (P_1, P_2, P_3, P_4, P_5) in the parameter space:

$$MOP1 = [x(T_f)/(x_0 * (1-n_f))]/[y(T_f)/(y_0 * (1-m_f))] \quad (5.4)$$
$$MOP2 = z(T_f)/z_0 \quad (5.5)$$
$$MOP3 = (Mop1.2 - V * T_f)/R_1 \quad (5.28)$$

5.8 EVOLUTION OF THE SYSTEM

5.8.1 Introduction

Up to this point, we have assumed the system to be at a basic operating point in the basic test matrix: for each basic operating point, a simplified model has been presented in order to apply the System Effectiveness Analysis methodology.

In this section, we define five basic operating points by assigning different values to the constants that remain to be fixed (Moe1, Moe3, $Mop3.7_{max}$ ...). Thus, we will have five basic operating points corresponding to four different sets of constants in the model. For each of these basic operating point, the excursion matrix will be explored by varying the parameters P_1, P_2, P_3, P_4, P_5.

5.8.2 Nominal performance

This is the basic operating point that corresponds to the nominal level of performance in the system; as defined in the IFFN documents [IFFN Test Plan, 1985], the nominal system includes all the components that can be removed to go from one test cell to another (Fig.5.3): in particular it includes the NATO airborne early warning (NE-3A), and the Joint Tactical Information distribution System (JTIDS). As mentioned in section 5.3, an unit refers to either an aircraft or a ground firing unit for the friend; it refers to an aircraft for the enemy as well as for the neutrals. We assume the following values for the constants of the model:

$V = 200$ meters per second

$x_0 = 100$ units

$y_0 = 150$ units
$z_0 = 50$ units
$b = 0.022$ units per second
$n_f = 0.6$
$m_f = 0.3$
$R_0 = 10000$ meters
$R_1 = 5000$ meters
Mop2.3 + Mop3.4 + Mop 5.3 = 7 seconds
$(\text{Mop3.7})_{max} = 10$ seconds
Moe1 = Moe3 = 1

5.8.3 Removal of the NE-3A

This basic operating point corresponds to the system defined in section 5.8.2 without the NATO airborne early warning system (NE-3A). We assume that the constants are the same as above except for the following ones:

$R_0 = 9000$ meters
Moe1 = Moe3 = 0.98

The removal of the NE-3A represents the removal of a source of information; without NE-3A, the system cannot monitor as large a detection volume as the nominal system; the removal of the NE-3A corresponds to a reduction in the detection and ID capabilities of the system, and therefore to a deterioration in the detection and ID performances.

5.8.4 Removal of the JTIDS

This basic operating point corresponds to the system defined in section 5.8.2 where the Joint Tactical Information Distribution System (JTIDS) is not present. We assume that the constants are the same as in section 5.8.2 except for the following ones:

Mop2.3 + Mop3.4 + Mop 5.3 = 8 seconds
$(\text{Mop3.7})_{max} = 12$ seconds
Moe1 = Moe3 = 0.98

The JTIDS is a high capacity information distribution system that provides secure, flexible and jam resistant information transfer among all the nodes of the system; its removal corresponds to increases in communication delays, and to less reliable detection and ID capabilities.

5.8.5 Removal of the JTIDS and of the NE-3A

This basic operating point corresponds to the system defined in section 5.8.2 where the JTIDS and the NE-3A have been removed. We assume that the constants are the same as in section 5.8.2 except for the following ones:

R_0 = 9000 meters
Mop2.3 + Mop3.4 + Mop 5.3 = 8 seconds
$(\text{Mop3.7})_{max}$ = 12 seconds
Moe1 = Moe3 = 0.95

These values are a combination of the ones assumed in sub-sections 5.8.3 and 5.8.4.

5.8.6 Removal of the JTIDS and inclusion of an additional NE-3A

This basic operating point corresponds to the nominal system where the JTIDS has been removed and where an additional NE-3A has been included in the system. This point is not defined in the IFFN documents [IFFN Test Plan 1985] and does not correspond to any point in the basic matrix of Fig.5.3; nevertheless this configuration will be considered later to apply the dynamic programming algorithm presented in Chapter 4. we assume that the constant are the same as in section 5.8.2 except for the following ones:

Mop2.3 + Mop3.4 + Mop 5.3 = 8 seconds
$(\text{Mop3.7})_{max}$ = 12 seconds
Moe1 = Moe3 = 1

The inclusion of an additional NE-3A into the system does not extend the detection volume. If one compares this configuration to the one defined in section 5.8.4 (nominal system without JTIDS), one should note that the inclusion of an additional NE-3A in the system only affects the probability of detection (Moe1) and the probability of identification

(Moe3): these two quantities are assumed to increase as an additional NE-3A gets included in the system.

In this section we have defined five basic operating points to which the SEA methodology will be applied.

5.9 CONCLUSION

In this chapter, an actual air defense system has been presented, and a simplified mathematical model of this system has been set up; this mathematical model will be the basis for the design of experiments to run on the test-bed. To this model (with a 5 dimensional parameter space and a 3 dimensional MOP space), we will apply the experiment design algorithm described in chapter 3; this algorithm will enable us to determine a small number of experiments to run on the test-bed, and then to reconstitute the system locus of the actual system/test-bed. To the actual system we will then apply the dynamic programming algorithm introduced in chapter 4.

The purpose of this example is to demonstrate that the methodologies introduced in previous chapters can be useful in solving a real engineering problem: evaluate experimentaly a system and improve it.

In chapter 6 we will present the results of the System Effectiveness Analysis applied to the IFFN simplified mathematical model described in this chapter.

Finaly, in chapter 7, the demonstration of the procedures introduced in chapter 3 and 4 will be carried out.

CHAPTER 6
RESULTS OF THE SYSTEM EFFECTIVENESS ANALYSIS METHODOLOGY APLLIED TO THE SIMPLIFIED IFFN MODEL

6.1 INTRODUCTION

The results of the System Effectiveness Analysis methodology applied to the mathematical model developed in chapter 5 are presented in this chapter. The SEA methodology will be applied to each of the five basic operating points defined in the previous chapter by varying parameters P_1 through P_5.

We will focus primarily on the basic operating point corresponding to the nominal level of performance of the IFFN system. Since the results are qualitatively the same for the four other operating points, they will be presented in less detail.

Plots are presented throughout this chapter: the data have been generated by solving the equations of the IFFN pròblem on a personal computer (IBM PC/AT). The graphics have been generated with a package [Bohner 1986] developed on the same machine, and plotted using an HP 7475A plotter.

6.2 MISSION LOCUS

6.2.1 Introduction

We assume that the mission the system has to perform is the same for the four basic operating points. Therefore, we only need to define one mission locus for the system.

6.2.2 Mission Requirements

As mentioned in chapter 2, in order to enable one to evaluate the system at hand

vis a vis the mission it has to perform, the mission requirements must be expressed in terms of the MOPs defined for the system.

Qualitatively, the mission the system has to fulfill is to deter enemy aircraft from invading friendly territory, without killing neutrals, and to prevent enemy aircraft from firing missiles aimed at friendly assets. In term of the MOPs defined in chapter 5 and summarized in Table 6.1, it means that MOP1 must be greater than 1, that MOP2 must be as close as possible to 1, and that MOP3 must be greater than 1.

Table 6.1: MOPs of the system

MOP1	compare the willingness of the two sides to keep on fighting. If MOP1>1, the friends are winning the battle
MOP2	Ratio of the neutrals not killed by friendly forces at the end of the battle to the initial number of neutrals
MOP3	Distance of the enemy's aircraft to FSCL at the end of the battle divided by the range of enemy's missiles

By definition, MOP2 is less than 1; let us show that MOP1 is less than $1/(1-n_f)$, and that MOP3 is less than R_0/R_1.

From the definition of MOP1, one can deduce

$$MOP1 = ((1-n)/(1-n_f))/((1-m)/(1-m_f)) \ . \tag{6.1}$$

Since $0 \leq n \leq n_f$ and $0 \leq m \leq m_f$, it follows that

$$MOP1 \leq 1/(1-n_f) \tag{6.2}$$

If MOP1 $= 1/(1-n_f)$, then the friendly forces are winning the battle without loosing

anything.

The range of variation of MOP3 is bounded too; indeed, since

$MOP3 = ((R_0 * P_2) - V * T_f)/R_1$, it follows that

$$MOP3 \leq R_0/R_1 \qquad (6.3)$$

The quantitative requirements are assumed to be as follows:

$$1/(1-n_f) \geq MOP1 > MOP1_0 = 1.1 \qquad (6.4)$$
$$1 \geq MOP2 \geq MOP2_0 = 0.8 \qquad (6.5)$$
$$1 \geq MOP3 \geq MOP3_0 = 1.0 \qquad (6.6)$$

Relation 6.4 requires the friends to win the battle with a 10% margin; inequality 6.5 requires that no more than 20% of neutrals be killed, and relation 6.6 requires the enemy forces to be stopped before they have crossed the line from which friendly positions are within range. These requirements define in the MOP space the mission locus shown in Fig.6.1. One should note that this mission locus is bounded.

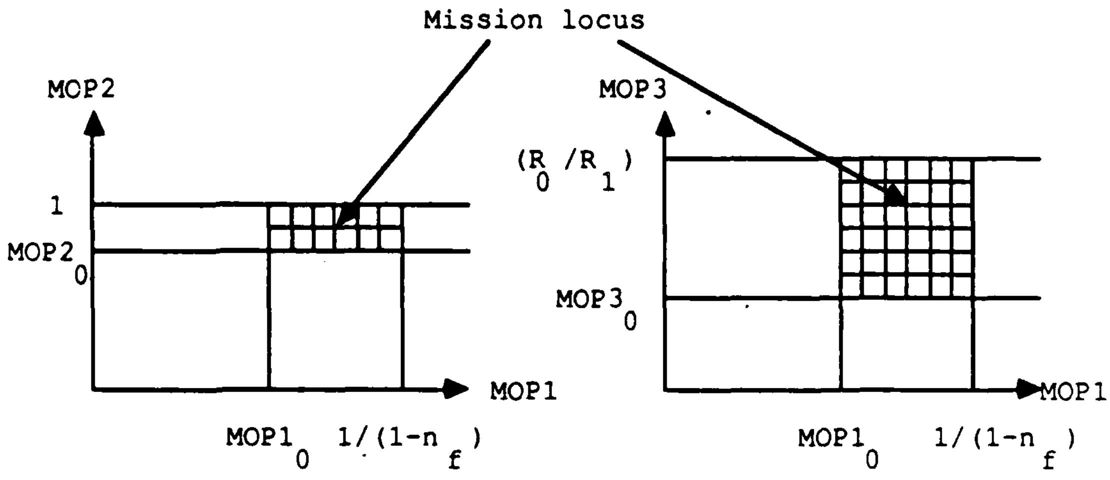


Fig.6.1: Projections of the Mission locus

For n_f=0.6, and the requirements set up above, if L_m denotes the mission locus

and $V(L_m)$ its volume, one can compute

$$V(L_m) = 0.28 \tag{6.7}$$

6.3 NOMINAL IFFN SYSTEM

6.3.1 Introduction

In the mathematical model that has been presented in chapter 5, five parameters and three MOPs have been defined: we are interested in ploting the system locus, that is the locus defined by the variation of the three MOPs as the five parameters are varied over their admissible ranges independently of each other.

The graphic package that has been used [Bohner 1986] draws a line between two MOP points if these two points correspond to the variation of one parameter at a time. Since we cannot vary the five parameters on a three dimensional picture, we will draw a sequence of volumes (partial loci) in the MOP space: for each of these volumes two parameters will be held constant, and the three other will be varied: the whole system locus will be defined by these partial loci as shown in Fig.6.2. For simplicity, only projections will be represented.

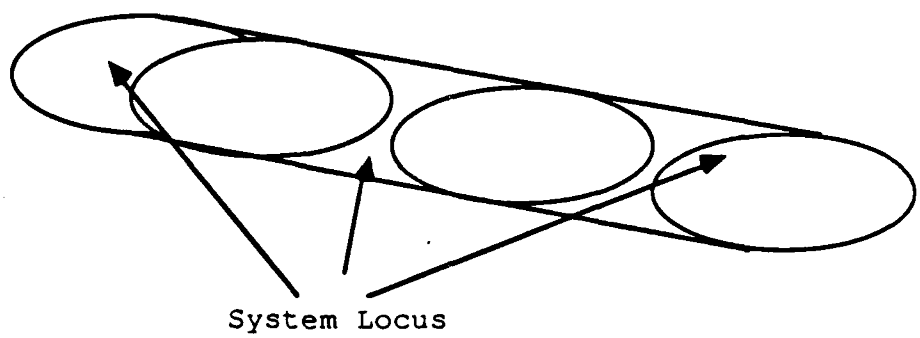


Fig.6.2: System locus defined by four partial loci

6.3.2 System Locus

To represent the system locus, as mentioned above, we will consider a family of partial loci: for each of these partial loci, parameters P_1 and P_2 will be held constant, and parameters P_3, P_4, P_5 will be varied. If P_1 is held constant, it means that the time delay to pass information from one node of the system to another is kept constant. Similarly, if P_2 is held constant, it means that the Air Control Procedure is not changed.

We assume the ranges of variation shown in Table.6.2 for the parameters:

Table 6.2: Parameters ranges

	Definition	Minimum	Maximum
P_1	Time delay to pass information	0.10	0.95
P_2	Air Control Procedure	0.75	0.95
P_3	Quality of identification	0.75	0.99
P_4	Level of centralization	0.50	0.99
P_5	Quality of Q&A IFF devices	0.75	0.99

These ranges have been chosen to yield realistic and physical ranges of variation for the aggregate quantities defined by Logicon [Logicon 1986](the "Mops" and "Moes" considered in chapter 5).

We will consider four partial loci, corresponding to the maximal and minimal values of P_1 and P_2 :

$P_1 = P_{1max}$ and $P_2 = P_{2max}$ -------> Partial Locus #1
$P_1 = P_{1min}$ and $P_2 = P_{2max}$ -------> Partial Locus #2
$P_1 = P_{1min}$ and $P_2 = P_{2min}$ -------> Partial Locus #3
$P_1 = P_{1max}$ and $P_2 = P_{2min}$ -------> Partial Locus #4

Before showing pictures of the whole locus, let us set P_1=constant, P_2=constant, P_3=constant and consider the set of MOP points obtained by varying P_4 and P_5: this will yield a "slice" of the partial loci we will obtain later, and give insight so as to how the locus looks like; a typical "slice" is shown in Fig.6.3 and Fig.6.4: Fig.6.3 corresponds to the projection of this slice on the plane (MOP1/MOP2), and Fig.6.4 on the projection on the plane (MOP1/MOP3).

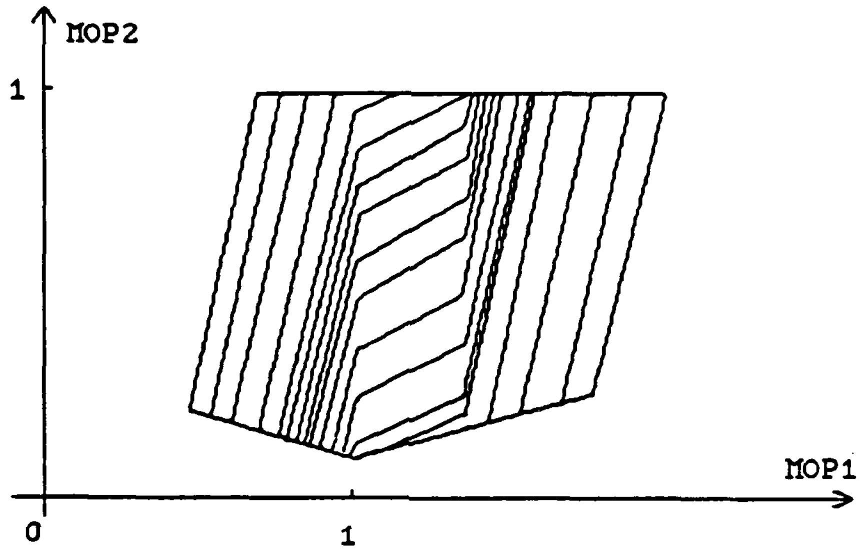


Fig.6.3: A slice of the system locus in the plane MOP1/MOP2

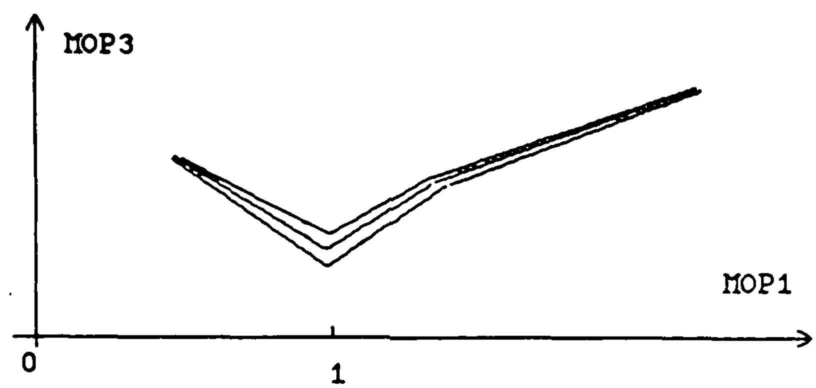


Fig.6.4: A slice of the system locus in the plane MOP1/MOP3

The purpose of these two figures is to show the shape of a typical slice of the system locus; actual and accurate plots will be shown later.

One can note an irregularity that coresponds to MOP1=1, that is to the change in the terminating condition (time T_f): if MOP1 is less than 1 the friends give up; if MOP1 is greater than 1, the enemies give up.

In Fig.6.4, MOP3 increases if MOP1 is greater than 1 and if MOP1 increases: indeed, the wider the margin by which the friendly forces are winning, the farther from the FSCL line the enemy is repulsed; on the other hand, if MOP1 is less than 1, MOP3 increases as MOP1 decreases: indeed, if the friends are loosing, the wider the margin by which they are loosing, the smaller the terminating time T_f is; the smaller the terminating time, the smaller the distance traveled by the enemy during the battle is. For example, if the friends give up immediately, the distance traveled by the enemy during the battle will be equal to zero; in this latter case, since the enemy aircraft are not repulsed, they will eventually invade the friend's territory.

In Fig.6.3, a vertical (or kinked) line is drawn for each value of P_4, that is for each level of centralization as represented by P_4 (P_4=1 for total centralization); the greater the value of P_4, the farther on the left of the diagram the corresponding vertical line is, and the smaller MOP1 is. It means that the lower the level of centralization, the greater MOP1 is, that is, the greater the chances of winning the battle are; this is the result of a trade-off between the accuracy in the ID process and the time needed to perform the identification: the more accurate the ID is, the longer it takes. It turns out that in the model, the time increase in the ID process due to a higher level of centralization is the most important of the two effects (the second effect being an increased accuracy). For a given vertical line (that is for P_4=constant), the greater P_5, that is the better the Question and Answer IFF devices are, the greater MOP2 is: it means that the better the Question and Answer devices, the greater MOP2 is, and the smaller the number of neutral killed is. These Q&A devices affect slightly MOP1 except around MOP1=1 where the quality of these devices is very important: around MOP1=1, the battle can be won or lost depending on the quality of the Q&A IFF devices.

For the next plots, P_3 will be varied with P_4 and P_5: we will obtain as many slices as the one of Figs.6.3 and 6.4, as values of P_3 considered. One should recall that P_3 represents the quality of identification (P_3=1 for perfect ID capabilities).

Projection on the plane MOP1/MOP2

In Fig.6.5, Fig.6.6, and Fig.6.7 projections on the plane (MOP1/MOP2) are represented; in Fig.6.5 the projection of partial loci #1 and #4 (their projections on the plane MOP1/MOP2 are the same) is shown, while in Fig.6.6 the projection of partial loci #2 and #3 (their projections on the plane MOP1/MOP2 are the same) is shown. For these two latter plots, if all parameters but P_3 are fixed, an increase in P_3 yields a higher MOP1 and a higher MOP2: the better the ID capabilities of the system the easier it is for the friends to win, and the smaller the number of neutrals killed by the system is.

Partial loci #1 and #4 correspond to $P_1=P_{1max}$, that is the longest time delay to pass information between two nodes of the system; on the other hand, partial loci #2 and #3 correspond to the shortest time delay to pass information between two nodes.From these two loci one can check the consistency of the model: the shorter the time to exchange information between nodes, the greater MOP1, and the greater the chances of winning the battle.

In Fig.6.7 the projection of the entire system locus on the plane MOP1/MOP2 is shown ; it is obtained by superposing the two previous plots.

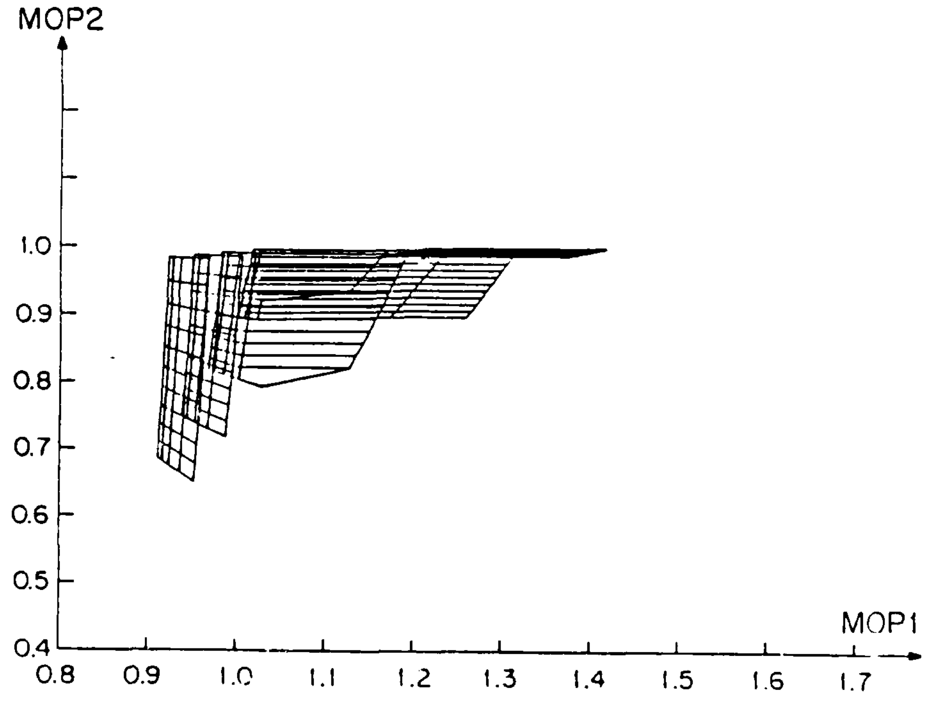


Fig.6.5: Partial Loci #1 and #4 projected on Plane MOP1/MOP2

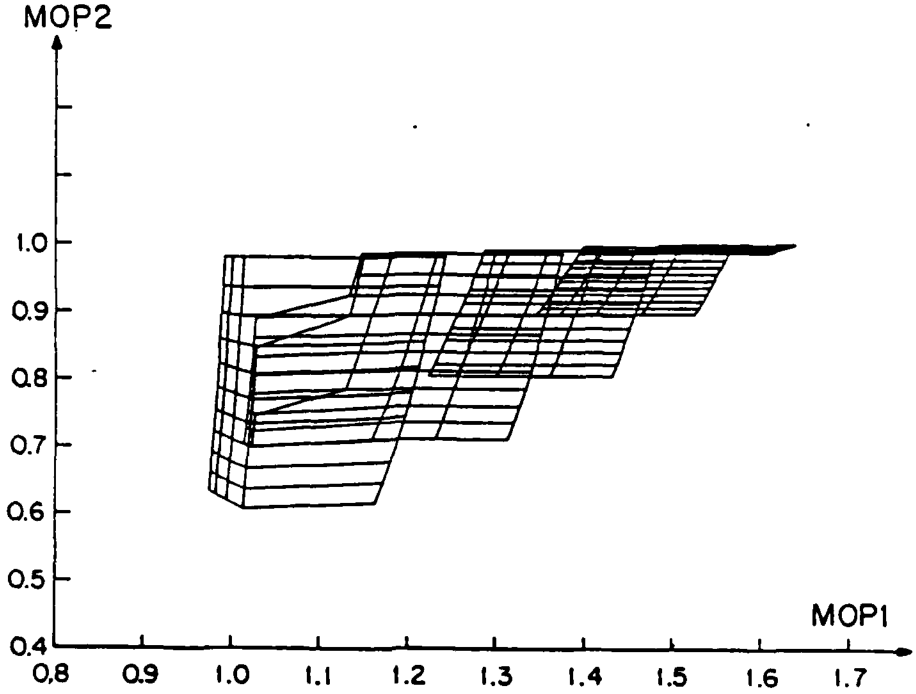


Fig.6.6: Partial Loci #2 and #3 projected on Plane MOP1/MOP2

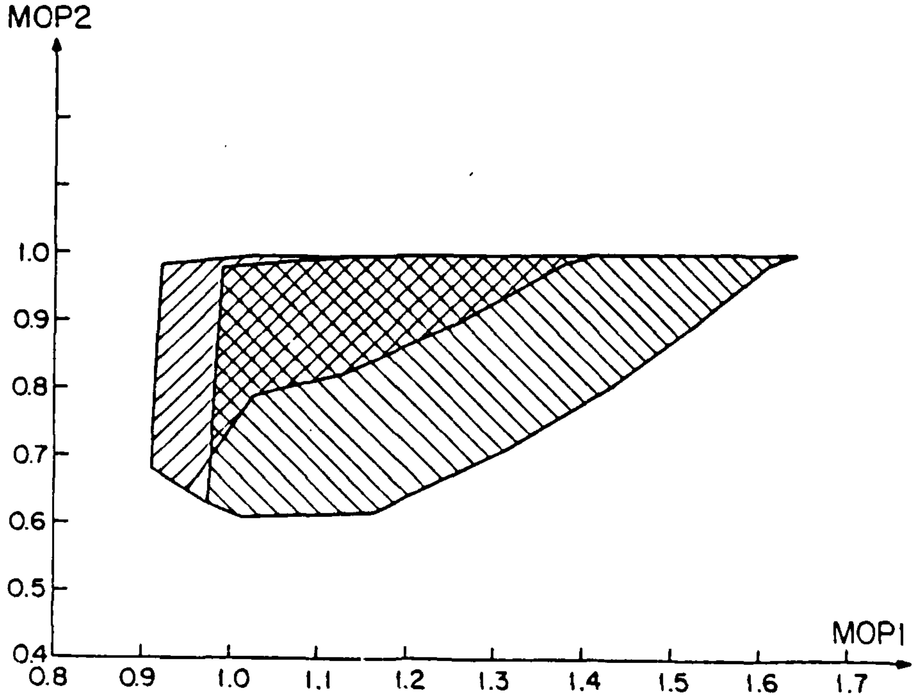


Fig.6.7: Entire System Locus projected on Plane MOP1/MOP2

Projection on the plane MOP1/MOP3

In Fig.6.8 the projections of partial loci #1 and #4 on the plane MOP1/MOP3 are shown . The upper part corresponds to partial locus #1, or to the highest quality of Air Control Procedure ($P_2 = P_{2max}$). The lower part of Fig.6.8 corresponds to partial locus #4 and to a low quality Air Control Procedure (ACP); the better the ACP , the greater MOP3: indeed, with a good ACP one can detect an enemy aircraft early and therefore stop it far away from the FSCL. The angle at MOP1=1 corresponds to the change in the terminating conditions; it corresponds to the irregularity already noted on previous plots around MOP1 = 1.

In Fig.6.9 the projections of partial loci #2 and #3 on the plane MOP1/MOP3 are shown . The upper part corresponds to partial locus # 2, and the lower part corresponds to partial locus #3; as in Fig.6.8, the better the ACP the greater MOP3 is.

In Fig.6.10 the projection of the entire locus defined as in Fig.6.3 is shown .

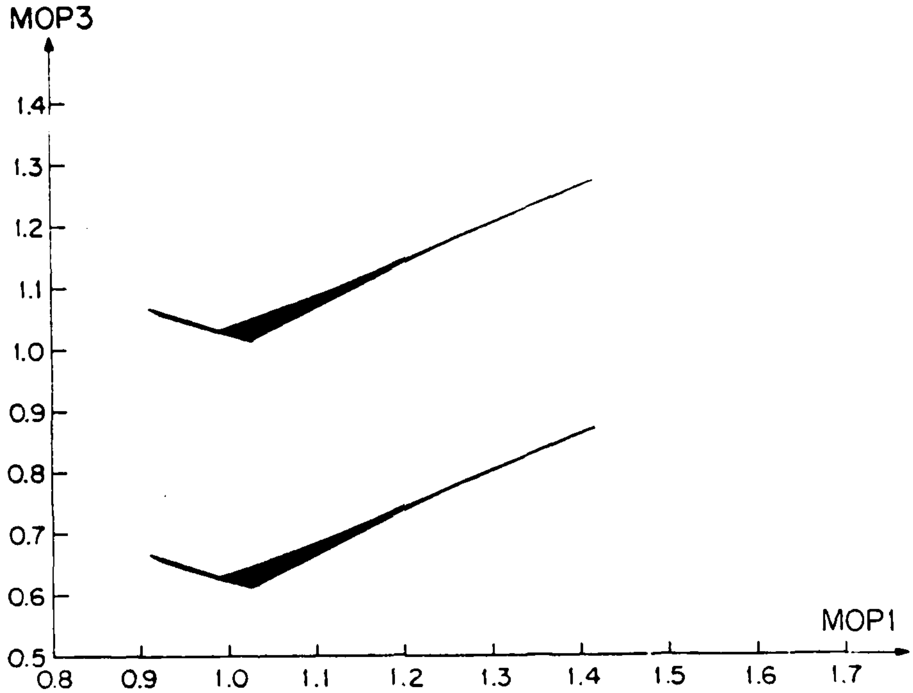


Fig.6.8: Partial Loci #1 and #4 projected on Plane MOP1/MOP3

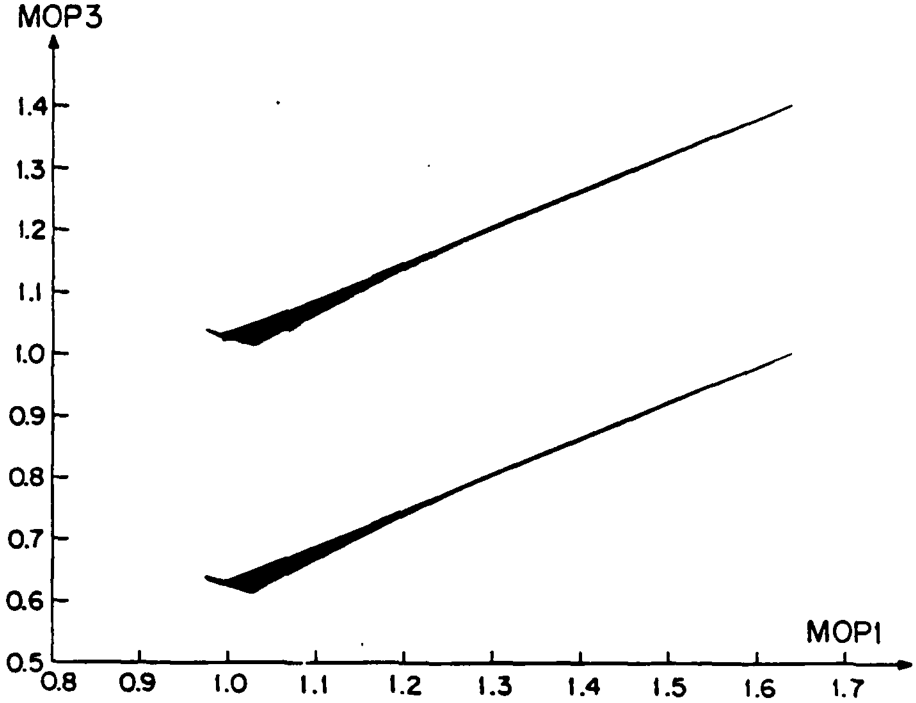


Fig.6.9: Partial Loci #2 and #3 projected on Plane MOP1/MOP3

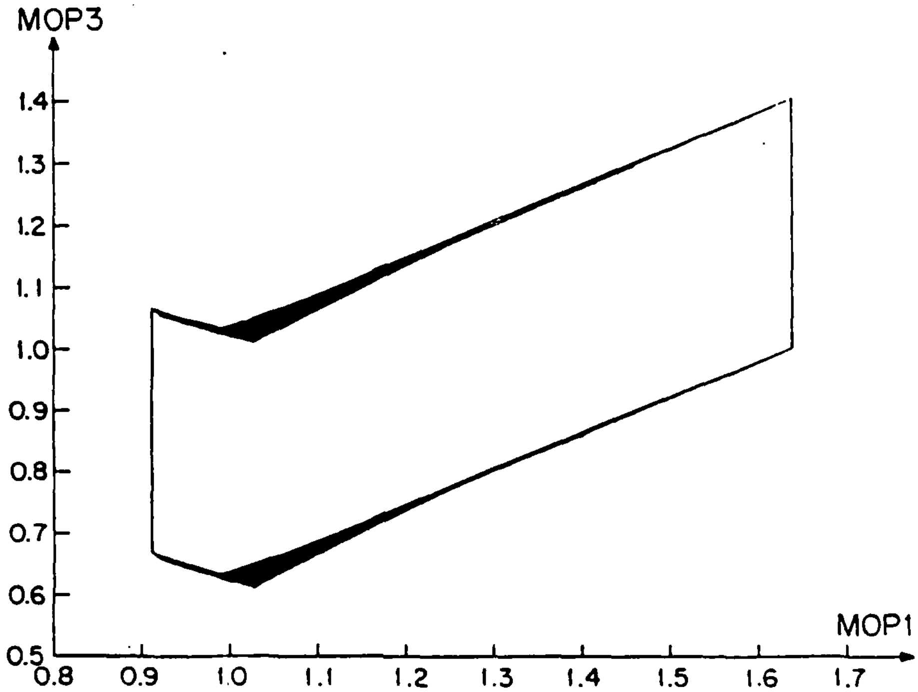


Fig.6.10: Entire System Locus projected on Plane MOP1/MOP3

Projection on the plane MOP3/MOP2

On Fig.6.11 partial loci #1 and #4 are shown ; partial locus #4 corresponds to the lowest quality of Air Control Procedure (ACP) and is on the left of the figure.

On Fig.6.12 partial loci #2 and #3 are shown ; partial locus #3 corresponds to the lowest quality of ACP and is on the left of the figure.

The projection of the entire system locus is determined by these four partial loci and is represented in Fig.6.13.

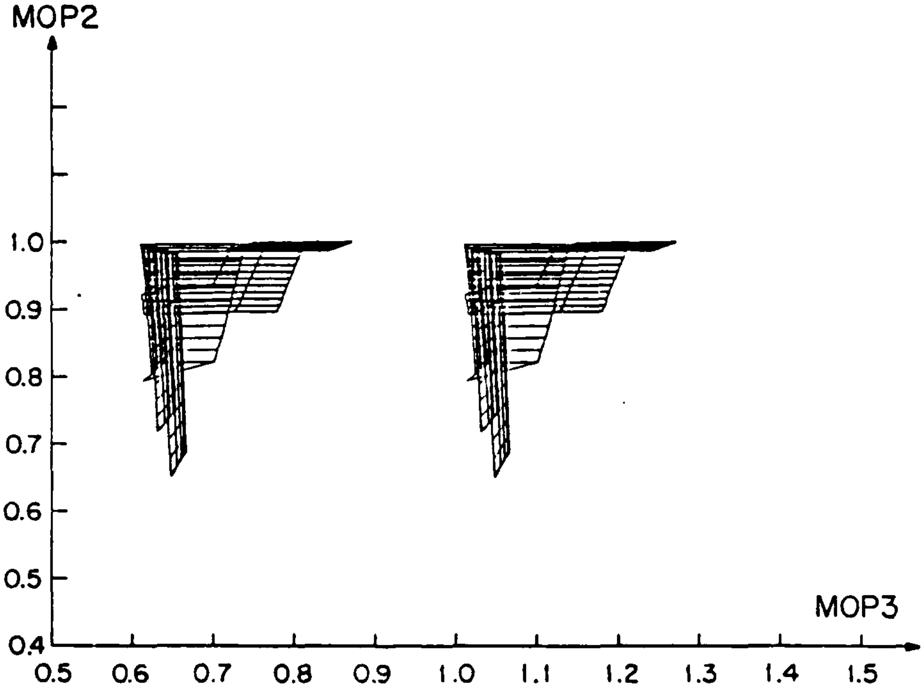


Fig.6.11: Partial Loci #1 and #4 projected on Plane MOP3/MOP2

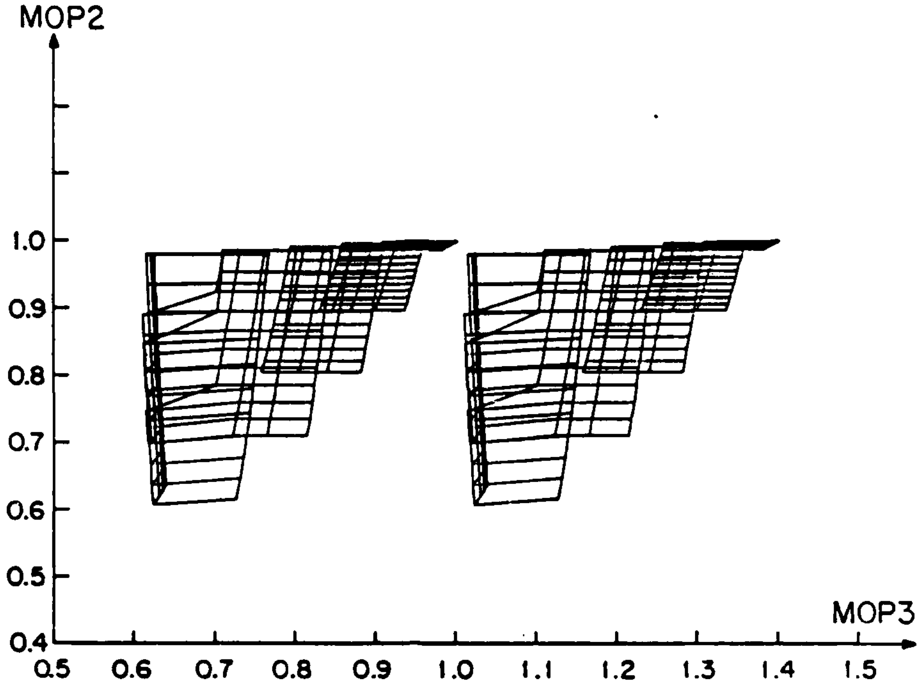


Fig.6.12: Partial Loci #2 and #3 projected on Plane MOP3/MOP2

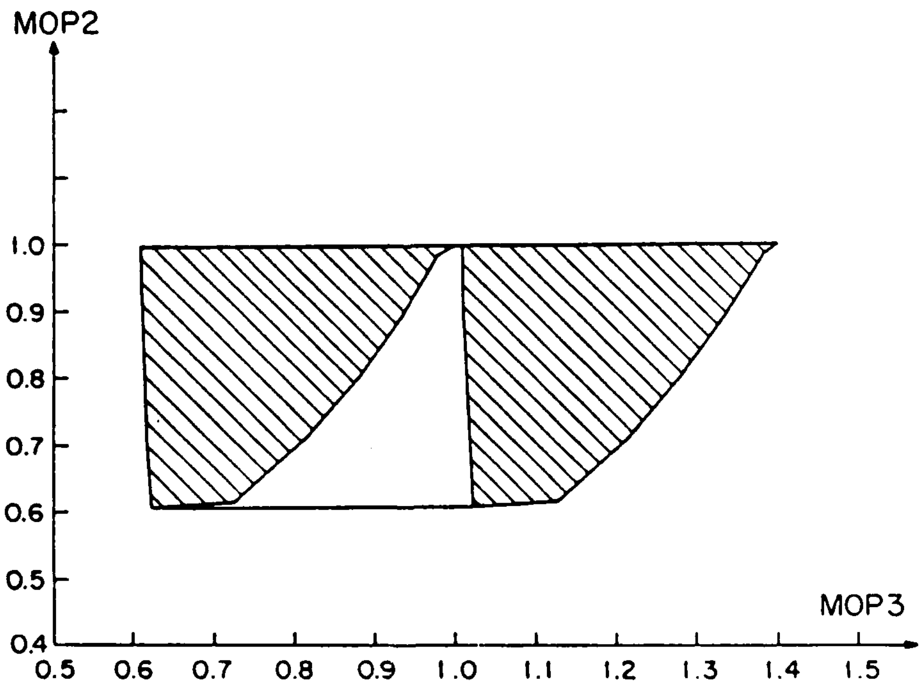


Fig.6.13: Entire System Locus projected on Plane MOP3/MOP2

6.3.3 Measures of Effectiveness

If L_s designates the system locus and L_m designates the mission locus, and if V(L) is the volume of L, then, to compute the effectiveness of the mathematical model of the system, one must evaluate $V(L_s \ll L_m)$, and $V(L_m)$ or $V(L_s)$ depending on the MOE one is interested in. With the definitions of chapter 2, E_1 will measure how well the system capabilities are used, E_2 will measure how well the mission is covered by the system, and E_3 will measure the degree of mismatch between system and mission.

For the basic operating point considered in this section, we have

$$V(L_s \cap L_m) = 0.020 \tag{6.12}$$
$$V(L_s) = 0.068 \tag{6.13}$$
$$V(L_m) = 0.28 \tag{6.6}$$

Therefore

$$E_1 = 0.292 \tag{6.14}$$
$$E_2 = 0.071 \tag{6.15}$$
$$E_3 = 0.061 \tag{6.16}$$

MOEs E_2 and E_3 are very small because of the size of the mission locus which takes into account such unrealistic events as the possibility for the friends of winning the battle without loosing any asset. Therefore, the degree of coverage of the mission locus by the system locus (E_2) is very low and the degree of mismatch ($1-E_3$) between the two loci is very high.

6.3.4 Conclusion

In this section the results of the System Effectiveness Analysis applied to the nominal system have been presented. Since the loci corresponding to other basic operating points are qualitatively the same as the ones shown above, in the next section of this chapter, only the numerical results of the SEA methodology will be presented.

6.4 NUMERICAL RESULTS FOR THE REMAINING BASIC OPERATING POINTS

6.4.1 Removal of the NE-3A

For this basic operating point defined in chapter 5 we have the following results

$$E_1 \approx 0.059 \tag{6.17}$$
$$E_2 \approx 0.011 \tag{6.18}$$
$$E_3 \approx 0.009 \tag{6.19}$$

Therefore, the removal of the NE-3A yields very significant reductions in the MOEs: the resulting configuration has an extremely poor effectiveness whatever the MOE considered is.

6.4.2 Removal of the JTIDS

For this basic operating point defined in chapter 5 we have the following results

$$E_1 \approx 0.113 \tag{6.20}$$
$$E_2 \approx 0.012 \tag{6.21}$$
$$E_3 \approx 0.011 \tag{6.22}$$

To the removal of the JTIDS corresponds smaller MOEs than in section 6.3.3, but the reductions are not as large as with the removal of the NE-3A; therefore, the NE-3A appears to be more important to the system than the JTIDS.

6.4.3 Removal of the NE-3A and of the JTIDS

For this basic operating point defined in chapter 5, the intersection of the system locus and the mission locus is the empty set (MOP3 is always less than 1); thus we have the following results

$$E_1 = 0 \tag{6.23}$$
$$E_2 = 0 \tag{6.24}$$
$$E_3 = 0 \tag{6.25}$$

6.4.4 Removal of the JTIDS and inclusion of an additional NE-3A

For this basic operating point defined in Chapter 5 we have the following results

$$E_1 = 0.167 \quad (6.26)$$
$$E_2 = 0.021 \quad (6.27)$$
$$E_3 = 0.019 \quad (6.28)$$

6.4.5 Summary of results

The results obtained in this chapter are summarized in Table 6.3.

Table 6.3: Summary of results

System configuration \ MOEs	E_1	E_2	E_3
Nominal	0.292	0.071	0.061
Without NE-3A	0.059	0.011	0.009
Without JTIDS	0.113	0.012	0.011
Without NE-3A and JTIDS	0	0	0
Additional NE-3A Without JTIDS	0.167	0.021	0.019

6.5 CONCLUSION

In this chapter, the System Effectiveness Analysis methodology has been applied to the mathematical model describing the IFFN system. The resulting measures are MOEs

for the simplified model.

In the next chapter, the real system will be evaluated by using the algorithm described in Chapter 3.

CHAPTER 7
APPLICATION: IMPROVEMENT OF AN ACTUAL SYSTEM

7.1 INTRODUCTION

In this chapter the methodology developed in chapter 3 and chapter 4 is applied to the IFFN system presented in chapter 5.

Five basic operating points have been defined for the IFFN system; we assume that the system at hand is the one corresponding to the fourth basic operating point: the Joint Tactical Information Distribution System (JTIDS) and the NATO airborne early warning (NE-3A) are not included in the initial system, and one is interested in determining the optimal sequence of modification for the system. These modifications can be: adding the JTIDS to the system, or adding the NE-3A. The optimal sequence of improvement will be determined for the actual system as opposed to the model presented in chapter 5: for each stage of the process, the methodology presented in chapter 3 will be used.

In the first part of this chapter, the effectiveness of the actual system at each basic operating point will be evaluated. The second part will deal with the improvement algorithm.

7.2 ACTUAL SYSTEM EFFECTIVENESS

7.2.1 Introduction

The results of the system effectiveness analysis will be presented extensively for the final system, that is the one obtained after adding the JTIDS and the NE-3A to the initial system. The reason for this choice is that one is interested mostly in the final system as opposed to any intermediate stage. For the four other basic operating points only numerical results will be presented.

7.2.2 Final system: actual system locus

The presentation of the results in this section follows the same format as in chapter 6 where the system locus of the mathematical model has been introduced. To determine the actual system locus, we will go through the flowchart of Fig 3.9.

Step 1: Determination of the model locus.

This step has been completed in chapter 6 where the results of the SEA methodology applied to the mathematical model of the IFFN system have been presented. The locus is shown in Figs. 6.5 to 6.13 in Chapter 6.

Step 2: Selection of points on the model locus.

As mentioned in chapter 3, we inscribe the model locus in a parallilepiped, and choose the points of contact between the model locus and the parallilepiped (or the center of gravity of these points). In what follows a row vector $\underline{x}$ is constructed with elements of three MOPs:

$$\underline{x} = [\ MOP1\ ,\ MOP2\ ,\ MOP3\ ] \tag{7.1}$$

For the locus obtained in chapter 6 and shown in Figs.6.5 to 6.13, six points of contact are obtained:

$$\underline{x}_{d1} = [\ 1.639\ ,\ 0.999\ ,\ 1.203] \tag{7.2}$$
$$\underline{x}_{d2} = [\ 1.432\ ,\ 0.999\ ,\ 1.081] \tag{7.3}$$
$$\underline{x}_{d3} = [\ 1.639\ ,\ 0.999\ ,\ 1.403] \tag{7.4}$$
$$\underline{x}_{d4} = [\ 0.911\ ,\ 0.687\ ,\ 0.867] \tag{7.5}$$
$$\underline{x}_{d5} = [\ 1.006\ ,\ 0.611\ ,\ 0.826] \tag{7.6}$$
$$\underline{x}_{d6} = [\ 1.028\ ,\ 0.987\ ,\ 0.610] \tag{7.7}$$

The six vectors obtained above represent the entire system locus as opposed to any of its region. The three first vectors correspond to maximum values for the MOPs in the system locus: $\underline{x}_{d1}$ corresponds to the maximum value of MOP1, $\underline{x}_{d2}$ corresponds to the maximum value of MOP2, and $\underline{x}_{31}$ corresponds to the maximum value of MOP3. These three first vectors are in the mission locus. One should note in Fig.6.7 that the number of contact points is infinite for MOP2=0.999: $\underline{x}_{d2}$ is defined as the center of gravity of this infinite number of points. Similarly in Fig.6.10, one can remark that the number of points

corresponding to the maximum value of MOP1 is infinite: $\underline{x}_{d1}$ is thus defined as the center of gravity of these points. The numerical values of the MOPs for $\underline{x}_{d1}$ can be interpreted as follows: MOP1=1.639 means that the friends are winning the battle with a margin of 64% or so: this is the largest possible margin for the system. MOP2=0.999 means that less than 1% of the neutrals are killed by the friendly forces. Lastly MOP3=1.203 means that the enemy forces are stopped before they have been able to fire any missiles.

The three last vectors correspond to minimum values for the MOPs in the system locus: $\underline{x}_{d4}$ corresponds to the minimum value of MOP1, $\underline{x}_{d5}$ corresponds to the minimum value of MOP2, and $\underline{x}_{d6}$ corresponds to the minimum value of MOP3. These three last vectors are not in the mission locus. One can note in Fig.6.10 that an infinite number of points correspond to the minimum value of MOP1: therefore, $\underline{x}_{d4}$ is defined as the center of gravity of these points. For this vector $\underline{x}_{d4}$, the friends loose the battle (MOP1<1), kill more than 30% of the neutrals and fail to prevent the enemy from firing missiles (MOP3<1).

Step 3: Inversion algorithm.

For each of the points $\underline{x}_{di}$ determined above, we compute a parameter value $\underline{p}_i$ such that $\underline{x}_{di} = f(\underline{p}_i)$, where f denotes the mathematical model of the system. If one notes a parameter vector $\underline{p}$ as a row vector

$$\underline{p} = [\, P_1 , P_2 , P_3 , P_4 , P_5 \,] \tag{7.8}$$

then, the parameter vectors corresponding to the $\underline{x}_{di}$ determined above are

$$\underline{p}_1 = [\, 0.205 , 0.850 , 0.990 , 0.500 , 0.904 \,] \tag{7.9}$$
$$\underline{p}_2 = [\, 0.554 , 0.850 , 0.990 , 0.694 , 0.911 \,] \tag{7.10}$$
$$\underline{p}_3 = [\, 0.204 , 0.950 , 0.990 , 0.500 , 0.900 \,] \tag{7.11}$$
$$\underline{p}_4 = [\, 0.685 , 0.850 , 0.750 , 0.846 , 0.933 \,] \tag{7.12}$$
$$\underline{p}_5 = [\, 0.521 , 0.851 , 0.821 , 0.652 , 0.750 \,] \tag{7.13}$$
$$\underline{p}_6 = [\, 0.873 , 0.750 , 0.874 , 0.990 , 0.990 \,] \tag{7.14}$$

These values were obtained using the algorithm described in Section 3.3 of Chapter 3. As expected, they are within the admissible range of variation in the parameter space as defined by Table 6.2.

The first parameter vector $\underline{p}_1$ corresponds to $\underline{x}_{d1}$ or to the maximum value for MOP1. This maximum value of MOP1 corresponds to the maximum margin by which the

battle is won. It is obtained for a small but not minimum time delay to pass information (first parameter), a medium quality of Air Control Procedure (second parameter), a maximum quality of identification (third parameter), a minimum level of centralization (fourth parameter), and a good (but not maximum) quality of Q&A IFF devices. Table 7.2 summarizes the physical significance of the parameter vectors obtained using the inversion algorithm.

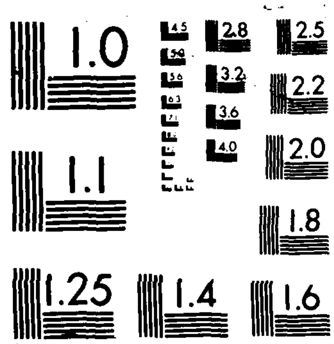
Table 7.1: Physical significance of the parameter vectors

		Time delay to pass Information	Quality of Air Control Procedure	Quality of Identification	Level of Centralization	Quality of Q&A IFF Devices
Maximum MOP1	$\underline{p}_1$	Small	Medium	Maximum	Minimum	High
Maximum MOP2	$\underline{p}_2$	Medium	Medium	Maximum	Medium	High
Maximum MOP3	$\underline{p}_3$	Small	Maximum	Maximum	Minimum	High
Minimum MOP1	$\underline{p}_4$	High	Medium	Minimum	High	High
Minimum MOP2	$\underline{p}_5$	Medium	Medium	Medium	Medium	Minimum
Minimum MOP3	$\underline{p}_6$	High	Minimum	Medium	Maximum	Maximum

MOP1 and the margin by which the battle is won is strongly linked to the quality of identification (ID): the maximum MOP1 is obtained for the maximum quality of ID and the minimum level of centralization, and the minimum MOP1 is obtained for the minimum quality of ID and the highest level of centralization. The fact that an increase in the quality of ID improves MOP1 is easy to predict. The greater the level of centralization, the lower MOP1 is: this is the result of a trade-off outlined in Chapter 6 between the increase in the accuracy of the ID process due to a higher level of centralization, and the increase in the time needed to perform this ID also due to a higher level of centralization: it turns out that the second of the two effects is the most important one, thus reducing MOP1. One should also note that MOP1 depends on the time delay to pass information between nodes: the

AD-A175 187 LARGE SCALE C3 SYSTEMS: EXPERIMENT DESIGN AND SYSTEM IMPROVEMENT(U) MASSACHUSETTS INST OF TECH CAMBRIDGE LAB FOR INFORMATION AND D P J MARTIN AUG 86 LIDS-TH-1580 2/2

UNCLASSIFIED N00014-85-K-0782 F/G 17/2 NL



CROCOPY RESOLUTION TEST CHART

smaller this time delay, the faster the response of the system, and the greater MOP1 is.

MOP2 appears to be linked to the quality of ID and to the quality of the Q&A IFFN devices which provide local ID information: the greater the quality of ID and the better the local ID information, the lower the number of neutrals killed by the friendly forces is.

MOP3 depends mostly on the Air Control Procedure (ACP), and on the time delay to pass information betweer nodes: the better the ACP and the smaller the time delay to pass information, the gr ...er MOP3 is. Indeed with a good ACP, one is able to detect the enemy far in the detection volume, and the smaller the time delay to pass information, the faster the response of the system and the greater MOP3 is.

Step 4: Experimental results.

At this stage, experiments are run at the parameter vectors determined at stage 3. Since we cannot run experiment on the actual system for the purpose of this thesis, a mathematical model which is slightly different from the one introduced in chapter 6 has been used. Let us introduce briefly this modified model that "represents" the actual system; this model is the same as the one of chapter 6 except that each time a parameter vector

$$\underline{p} = [\, P_1 , P_2 , P_3 , P_4 , P_5 \,]$$

is used in an equation, this parameter $\underline{p}$ is replaced by

$$\underline{p}' = [\, P'_1 , P'_2 , P'_3 , P'_4 , P'_5 \,]$$

with, $$P'_i = P_i * (1 + (P_i - P_{i0}) * \|\underline{p} - \underline{p}_0\|) \tag{7.15}$$

Therefore, the parameter values in the modified model are affected by the distance of the parameter vector considered to the parameter vector $\underline{p}_0$ introduced in chapter 3; for the case at hand we choose

$$\underline{p}_0 = [0.6 , 0.85 , 0.87 , 0.75 , 0.87 \,] \tag{7.16}$$

This vector has been selected because it is a central point in the parameter locus; the simplified model is calibrated at this point since for $\underline{p} = \underline{p}_0$, and the model and the actual system (as represented by the modified model) coincide.

The pseudo-experimental values obtained by exercising the modified model are

$$\underline{x}_{e1} = [\ 1.513\ ,\ 0.999\ ,\ 1.128\ ] \tag{7.17}$$
$$\underline{x}_{e2} = [\ 1.436\ ,\ 0.999\ ,\ 1.082\ ] \tag{7.18}$$
$$\underline{x}_{e3} = [\ 1.509\ ,\ 0.999\ ,\ 1.372\ ] \tag{7.19}$$
$$\underline{x}_{e4} = [\ 0.911\ ,\ 0.688\ ,\ 0.868\ ] \tag{7.20}$$
$$\underline{x}_{e5} = [\ 0.997\ ,\ 0.617\ ,\ 0.830\ ] \tag{7.21}$$
$$\underline{x}_{e6} = [\ 1.006\ ,\ 0.903\ ,\ 0.565\ ] \tag{7.22}$$

Step 6: Transformation from the model locus into the actual locus.

If $\underline{x}_m$ is a point in the model locus and if $\underline{x}_a$ is the corresponding point in the actual locus, the least square procedure presented in chapter 3 yields the following transformation

$$\underline{x}_a = T(\underline{x}_m) = L\underline{x}_m + \underline{V}\ , \tag{7.23}$$

where L is a linear transformation (defined by a 3x3 matrix) and $\underline{V}$ a constant translation vector. For the example at hand we have

$$L = \begin{bmatrix} 0.831 & 0.046 & -0.151 \\ 0.042 & 0.813 & 0.0008 \\ -0.007 & 0.078 & 1.127 \end{bmatrix} \qquad \underline{V} = \begin{bmatrix} 0.142 \\ 0.017 \\ 0.043 \end{bmatrix} \tag{7.24}$$

Step 7: Construction of the actual locus.

For each point in the parameter locus we apply A = Tof where "o" denotes the composition of two functions and where "f" stands for the mathematical function that maps the parameter locus into the model system locus.

Since we have considered the actual system to be "represented" by a modified model which has the same structure as the simplified one, the actual locus and the model locus have the same shape and the same structure. If real experiments could be run, the actual locus might be very differen from the model locus.

For the example at hand, since the shape of the actual locus in qualitatively the same as the one of the model locus, only comparisons of the two loci will be presented: in the following plots, the contours of the projections of both the model and the actual locus are shown; these projections are done on the planes MOP1/MOP2 (Fig.7.1), MOP1/MOP3 (Fig.7.2), MOP2/MOP3 (Fig.7.3).

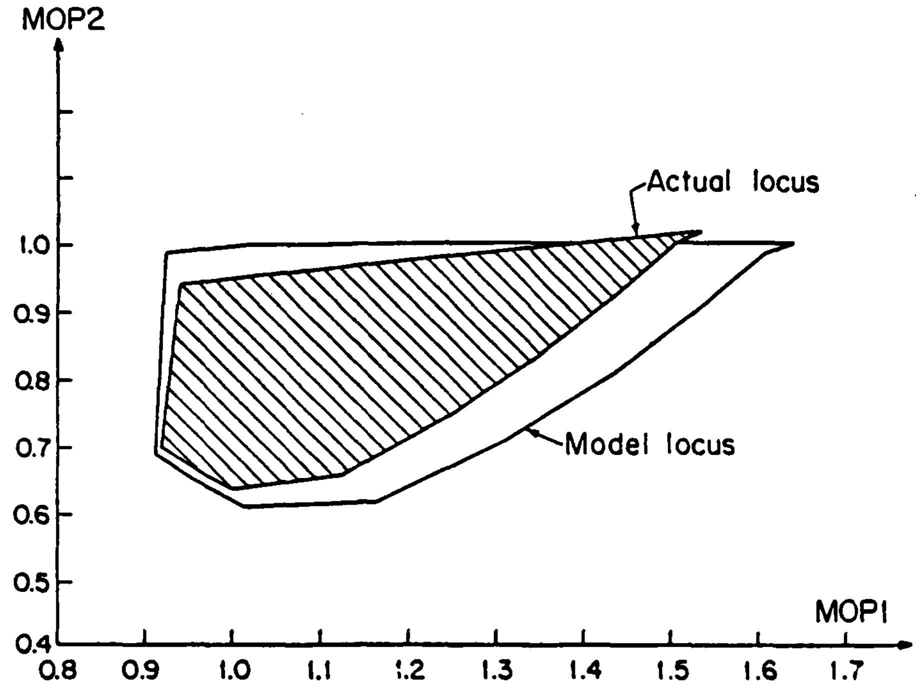


Fig.7.1: Actual and Model Loci projected on Plane MOP1/MOP2

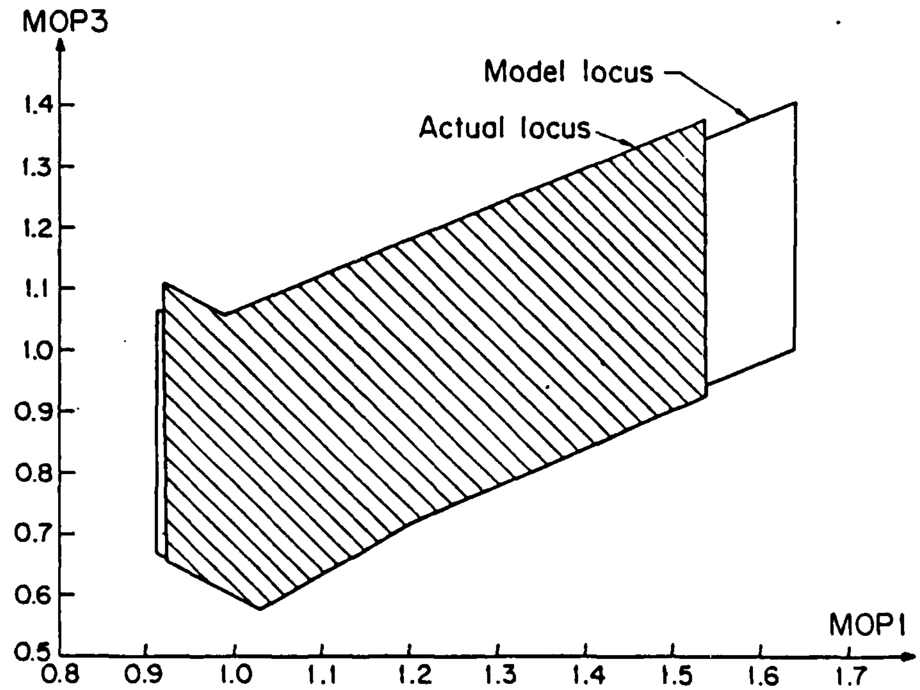


Fig.7.2: Actual and Model Loci projected on Plane MOP1/MOP3

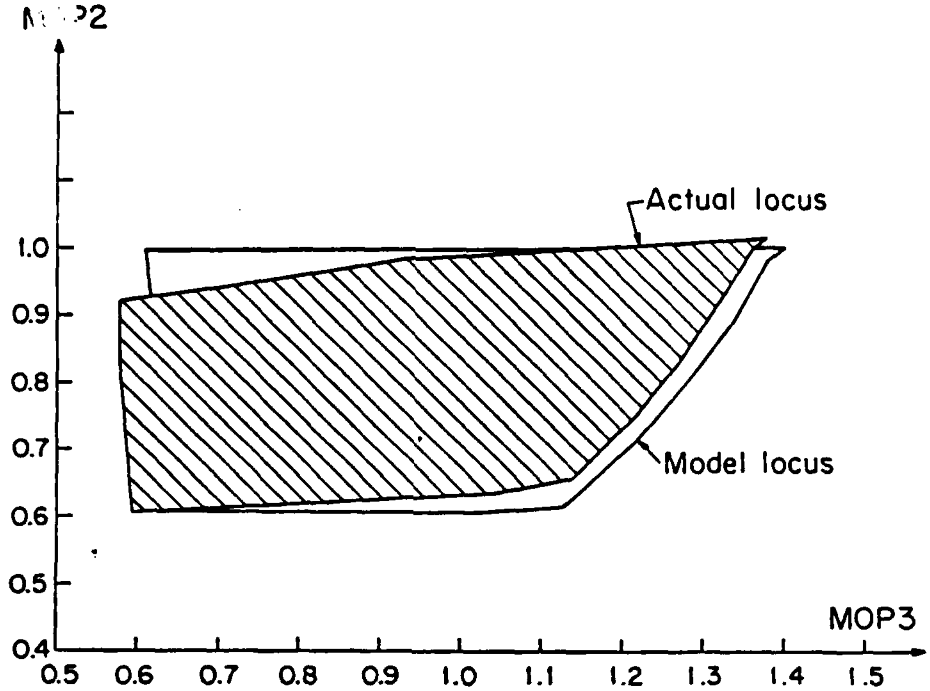


Fig.7.3: Actual and Model Loci projected on Plane MOP3/MOP2

7.2.3 Effectiveness of the actual system

From the actual locus of the nominal system constructed in section 7.2.2, one can evaluate the effectiveness of the system. If L_{sa} denotes the actual system locus, then with the notation of Chapter 2, the measures of effectiveness for the actual system are:

$$E_1 = V(L_{sa} \cap L_m)/V(L_{sa}) \tag{7.25}$$

$$E_2 = V(L_{sa} \cap L_m)/V(L_m) \tag{7.26}$$

$$E_3 = V(L_{sa} \cap L_m)/V(L_{sa} \cup L_m) \tag{7.27}$$

For the nominal system we have

$$E_1 = 0.200 \quad , \quad E_2 = 0.162 \quad , \quad E_3 = 0.152$$

Only numerical results are presented for the remaining four basic operating points. In Table 7.2 the results of the effectiveness analysis are presented for the five configurations of the actual system.

Table 7.2: Effectiveness of the actual system

Actual System configuration \ MOEs	E_1	E_2	E_3
Nominal	0.300	0.049	0.044
Without NE-3A	0.075	0.006	0.005
Without JTIDS	0.172	0.006	0.006
Without NE-3A and JTIDS	0.022	0.002	0.002
Additional NE-3A Without JTIDS	0.200	0.162	0.152

In order to enable one to compare the results obtained for the model and the ones obtained for the actual system Table 6.3 of Chapter 6 is reproduced in Table 7.3.

Table 7.3: Effectiveness based on the model

System configuration \ MOEs	E_1	E_2	E_3
Nominal	0.292	0.071	0.061
Without NE-3A	0.059	0.011	0.009
Without JTIDS	0.113	0.012	0.011
Without NE-3A and JTIDS	0	0	0
Additional NE-3A Without JTIDS	0.167	0.021	0.019

The values of E_1 obtained for the actual system (Table 7.2) are slightly greater

than the ones obtained for the model (Table 7.3); in particular, the effectiveness E_1 of the actual system without NE-3A and JTIDS is very small, but not equal to zero. It means that the capabilities of the actual system are better used than one could have thought by studying the model only. On the other hand, for the first three basic operating points, the values of E_2 and E_3 are slightly smaller for the actual system than they are for the model: the degree of coverage of the mission is smaller for the actual system than for the model, and the degree of misfit between mission and system loci is greater for the actual system than for the model. For the last two basic operating points, the degree of coverage of the mission is smaller for the actual system than for the model, and the degree of misfit between mission and system loci is greater for the actual system than for the model.

7.3 SYSTEM PLANNING

7.3.1 Assumptions

In this section, the dynamic programming algorithm introduced in Chapter 4 is applied to the "actual system". We assume that the initial system is the one without NATO Airborne Early Warning (NE-3A) and Joint Tactical Information Distribution System (JTIDS) and we consider a two stage evolution: the initial system is denoted by X_0 ; at stage 1, the system is X_1, and at stage 2 it is X_2. At each stage, one can add either the JTIDS or the NE-3A but not both. Therefore the set of possible decisions is assumed to be known. Another assumption is that the only MOE of interest for the developer is E_1: it means that the developer is interested in knowing how well the capabilities of the system are used. The goal of the evolution process is the maximization of

$$V = E_1(1) + E_1(2) \tag{7.28}$$

where $E_1(i)$ stands for E_1 at stage i.

With the set of possible decisions considered in this section, it is possible for the final system to be composed of the initial system X_0 to which have been added two JTIDS: this doesn't make any sense, and this possibility should be eliminated. The branch leading to this particular state will not be considered in the decision tree to be sketched as in Chapter 4.

7.3.2 Application

With these assumptions and the framework provided in Chapter 4, one can draw a decision tree similar to the one of Fig.4.3. In Fig.7.4 this decision tree is shown: at each stage, the upper branch corresponds to the addition of a NE-3A, and the lower branch corresponds to the addition of a JTIDS.

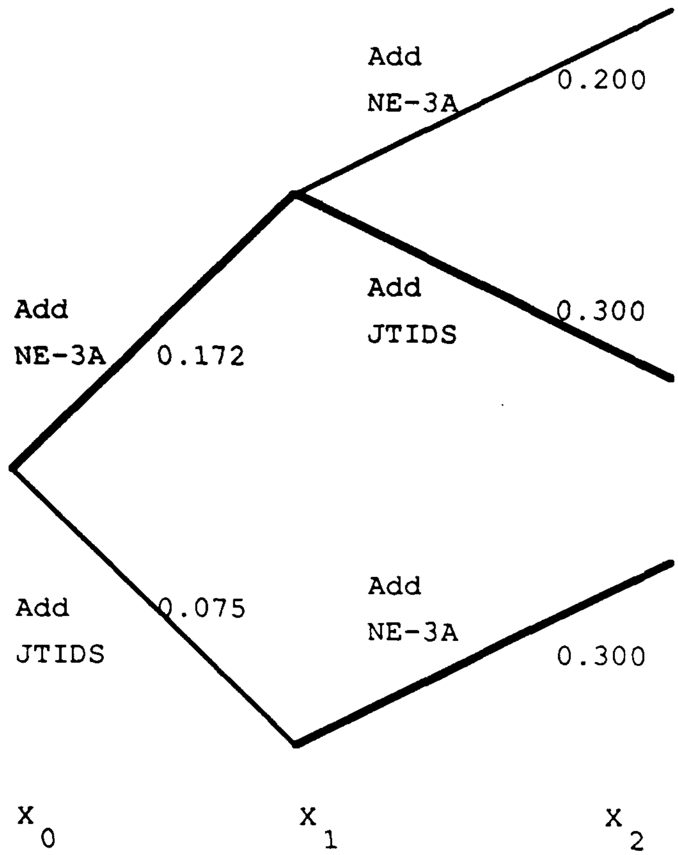


Fig.7.4: Optimal evolution sequence for the IFFN system

From Fig.7.4, one can deduce the optimal sequence of improvements for the initial system X_0: first add to the initial system the NE-3A, to yield X_1; second add the JTIDS to X_1 to yield the final system. The optimal value of V defined by equation 7.28 is then

$$V = 0.472 \quad . \tag{7.29}$$

The dynamic programming algorithm introduced in Chapter 4 has been used for the actual IFFN system whose effectiveness has been analysed in section 7.2. It allows one to plan the evolution of the system: in the IFFN example, the optimal sequence of modification is. First, add an early warning component (NE-3A) whose function is to increase the detection volume of the system and to increase the probabilities of detecting and identifying an aircraft. Second, add a high capacity information distribution system (JTIDS) whose function is to provide faster communication between the nodes of the system, and to increase the probabilities of detecting and identifying an aircraft because of higher communication reliability. Therefore, for the initial system, increasing the detection volume is the first priority. The second priority is the addition of a component providing the system with fast communication capabilities.

7.4 CONCLUSION

In this chapter, the methodology developed in Chapter 3 and Chapter 4 has been applied to the IFFN system presented in Chapter 5: the procedure to evaluate the effectiveness of an actual system has been demonstrated as well as the system planning procedure.

The methodology applied in this chapter provides the system developer with a powerful tool since it enables him to make the best possible use of a test bed: it allows him to assess the effectiveness of different configurations of the system by running the smallest possible number of experiment; it also allows him to determine an optimal sequence of improvement as the system goes from one configuration to another.

The next chapter concludes this thesis and gives directions for further research.

CHAPTER 8
CONCLUSIONS AND DIRECTIONS FOR FURTHER RESEARCH

8.1 CONCLUSIONS

As pointed out in Chapter 1, since a large scale system often cannot be exercised to obtain needed performance data, one must build a test bed. Such a test bed provides a means to gather data on the system; it also provides a means for simulating different configurations of the system. In this thesis, a methodology to make the best possible use of a test bed has been presented.

The first step in the methodology aims at determining the smallest number of experiments that need to be run to evaluate the effectiveness of an actual system. A basic assumption is that a simplified mathematical model of the system at hand can be obtained. This model is used to determine experiments; then it is used together with experimental results to yield the effectiveness of the system. This first step provides the system developer with a powerful tool since it enables him to select and design a very small number of experiments to run on the test bed in order to evaluate its effectiveness.

In the second step, a system planning procedure for optimizing the evolution of a large scale system is used. After evaluating the effectiveness of different configurations of the system with the first step, this procedure enables the system developer to select the best evolution path for the system at hand. The basic assumption underlying this procedure is that the possible improvements are constrained to belong to a fixed set of improvements.

This methodology has been illustrated by applying it to an air defense system currently under development, and known as the "Identification Friend Foe Neutral system", or "IFFN system". Four different configurations of this system have been considered, and the effectiveness of the system in each of these configurations has been evaluated. Then, the system planning procedure has been applied to these four configurations on the basis of the effectiveness evaluation, to determine an optimal sequence of modifications for the system.

8.2 DIRECTIONS FOR FURTHER RESEARCH

Continuation of this research should proceed as the methodology gets applied to new examples. In this section, recommendations for future research are given; they focus on the definition of boundaries for a system, and on the optimization of a system's evolution.

Physical and temporal boundaries of a system: The definitions and guidelines provided in this thesis to draw the boundaries of a system should be further tested on actual examples and expanded, if necessary. In particular, the notions of evolving system and of temporal boundary for a system should be further developed.

Optimal evolution of a system: Further research to expand the framework of the dynamic programming algorithm presented in this thesis will be helpful. Indeed, a basic assumption underlying the algorithm is that the set of decisions that can be taken by the system developer is known at the planning stage of the development. This assumption is justified for systems that cannot be built at once because of time and budget constraints. On the other hand, if a system is to integrate at each stage of its evolution the technological state of the art, one cannot assume that the set of all possible improvements will be known at the planning stage. For such systems, relaxing this particular assumption is crucial, if one wants to optimize their evolution.

REFERENCES

Bohner C.M. (1986), "Computer Graphics for System Effectiveness Analysis," MS Thesis, LIDS-TH-1573 , Laboratory for Information and Decision Systems, MIT, Cambridge, MA.

Bouthonnier V. and Levis A.H. (1984) , "Effectiveness Analysis of C^3 Systems," IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-14, No. 1.

Cothier P.H. (1984), "Assessment of Timeliness in Command and Control," MS Thesis, LIDS-TH-1391, Laboratory for Information and Decision Systems, MIT, Cambridge, MA.

Cothier P.H. and Levis A.H. (1985), "Assessment of Timeliness in Command and Control," Proc. 8th MIT/ONR Workshop on C^3 Systems, LIDS-R-1519, Laboratory for Information and Decision Systems, MIT, Cambridge, MA.

Debreu G. (1968), Theory of Value: An Axiomatic Analysis of Economic Equilibrium, Wiley, New York.

Dersin P. and A.H. Levis (1981), Large Scale Systems Effectiveness Analysis. Report LIDS-FR-1072, Laboratory for Information and Decision Systems, MIT, Cambridge, MA.

Dockery J.T. (1985), Theory of Effectiveness Measures. Technical Memorandum STC TM-729, SHAPE Technical Centre, The Hague, Netherlands.

Ekchian L.K. (1982). "An Overview of Lanchester-Type Combat Models for Warfare Scenarios". LIDS-P-1193, Laboratory for Information and Decision Systems, MIT, Cambridge, MA.

Gandee P.L. (1986), "Evaluation Methodology for Air Defense Command and Control System", MS Thesis, Naval Postgraduate School, Monterey, CA .

Karam J.G. (1985),"Effectiveness Analysis of Evolving Systems," MS Thesis, LIDS-TH-1431, Laboratory for Information and Decision Systems, MIT, Cambridge, MA.

Karam J.G. and Levis A.H. (1985), "Effectiveness Assessment of the METANET Demonstration," Proc. 7th MIT/ONR Workshop on C^3 Systems, LIDS-R-1437, Laboratory for Information and Decision Systems, MIT, Cambridge, MA.

Mishan E.J. (1976), Cost-Benefit Analysis, Praeger Publishers, New York.

Moose P.H., Wozencraft J.M. (1983), Lanchester Equations and Game Theory. Proceedings of the 6th MIT/ONR Workshop on C^3 systems, MIT, Cambridge, MA.

Taylor J.G. (1974), Lanchester-Type Models of Warfare and Optimal Control. Naval Research Logistics Quarterly, Vol.21, No.1, pp.79-106.

White J.A., Agee M.H., Case K.E. (1985), Principles of Engineering Economic Analysis, Wiley, New York.

Identification Friend, Foe, or Neutral (IFFN). Test Plan (TP), (1985), Air Force Army Joint Test Force, November.

Logicon (1986), MOE/MOP Mapping to Air Defense Functions, document no 85-0121, Logicon, Inc., Tactical and Training Systems Division, San Diego, CA, February.

Logicon (1986), MOE/MOP Mapping to Air Defense Functions, document no. 09-0124, Logicon, Inc., Tactical and Training Systems Division, San Diego, CA, April.

END

2-87

DTIC